01 Jan 2005

# A Measure of Robustness Against Multiple Kinds of Perturbations

Shoukat Ali
*Missouri University of Science and Technology*

Behdis Eslamnour



## Recommended Citation

S. Ali and B. Eslamnour, "A Measure of Robustness Against Multiple Kinds of Perturbations," *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2005.
The definitive version is available at https://doi.org/10.1109/IPDPS.2005.43

# A Measure of Robustness Against Multiple Kinds of Perturbations

Behdis Eslamnour and Shoukat Ali
Department of Electrical and Computer Engineering
University of Missouri-Rolla
Rolla, MO 65409-0040 USA
{ben88, shoukat}@umr.edu

## Abstract

*Parallel and distributed heterogeneous computing systems may operate in an environment that undergoes unpredictable changes causing certain system performance features to degrade. Such systems need robustness to guarantee limited degradation despite fluctuations in the behavior of its component parts or environment. Our previous work in this area presented a method for generating a measure of robustness for a given system. However, the focus of that approach was on a scenario where all perturbations were of the same kind, e.g., all perturbations were in message sizes or computation times, but not both message sizes and computation times. This paper gives an extended discussion of the case where perturbations could be of different kinds, and presents some new insights.*

**Keywords**: robustness, robustness metric, resource allocation, resource management systems, parallel and distributed systems.

## 1. Introduction

The robust design of computing and communication systems is becoming an increasingly important issue [1, 3–15]. There is a need for research that addresses the issue of developing a generalized robustness metric. In this paper, we extend our previous formulation ( [2]) of a standard generalized robustness metric for resource allocation. This will be an important step towards ongoing efforts to create robust designs.

The motivation for this research was provided by research supported by the DARPA's ITO Quorum program, under the project called "Management System for Heterogeneous Networks." The research involved the design and analysis of heuristics for robust resource allocation in different types of heterogeneous computing environments including the HiPer-D (High Performance Distributed Computing Program). A typical HiPer-D computing system consists of a set of dedicated machines interconnected by high-speed communication links. A set of sensors (radars, sonars, etc.) sends streams of data sets to a set of communicating, continuously running applications that process these data sets and send their outputs to other applications or actuators.

A HiPer-D system is required to satisfy a set of throughput and latency constraints. Any allocation of the resources must enforce these quality of service (*QoS*) constraints by ensuring that the computation and communication times are within certain limits. When the system is first configured, it is assumed to operate under certain estimated values of the initial sensor loads (i.e., outputs from sensors). Such an initial resource allocation ensures that all throughput and latency constraints are met when the system is first deployed. However, the system is expected to operate in a dynamic environment, where the sensor loads are expected to change unpredictably. Increases in sensor loads cause increases in the computation and communication times, which in turn may cause throughput and latency violations. Therefore, the initial resource allocation might be rendered invalid soon after the operation begins.

One way of handling the unpredictable load increases is to design a resource allocation that will tolerate as much sensor load increase as possible before a QoS violation occurs. For such an approach, how does one determine which resource allocation tolerates the largest load increase, given a set of resource allocations? This task necessitates the formulation of an appropriate metric.

One needs a general approach because the sensor loads might not be the only uncertainties in a HiPer-D system. Two other examples are: (a) inaccurate models for computation/communication times, and (b) sudden machine or link failures. A general approach is necessary also because for systems other than HiPer-D, there might be other uncertainties. Typically, the resource allocation decisions and the performance prediction are based on estimated/initial values of application and system parameters. However, complex computing and communication systems typically operate in an unpredictable environment where the actual values of these parameters may differ from the estimates due to a variety of reasons. As a result, the "real" system performance may degrade. An important question then arises. *Given a resource allocation, what is the maximum departure from the expected conditions that the system can tolerate and still deliver the promised performance?* That is, how *robust* is the system? Our research in [2] presented a method for generating a measure of robustness for a given system. However, the focus of that approach was on a scenario where all perturbations were of the same kind, e.g., all perturbations were in message sizes or computation times, but not both message sizes and computation times. The research in [2] did outline a method for generating a robustness measure when the perturbations were of mixed kinds. This paper gives an extended discussion of that method, and presents some new insights.

The rest of the paper is organized as follows. Section 2 summarizes the FePIA procedure given in [2] for generating a robustness measure for an arbitrary system. Section 3 presents our extension of the work in [2] to the scenarios where there are multiple perturbation parameters of different kinds. Section 4 concludes this paper. A glossary of the notation used in this paper is given in Table 1.

## 2. A Method for Generating Robustness Metrics

The research in [2] proposed a general procedure, called *FePIA,* for deriving a general robustness metric for any desired computing environment. For reference, we summarize the FePIA procedure here. Please see [2] for details.

1) Describe quantitatively the requirement that makes the system robust. Based on this *robustness requirement*, determine the QoS performance features that should be limited in variation to ensure that the robustness requirement is met. Identify the acceptable variation for these feature values as a result of uncertainties in system parameters. Mathematically, let $\Phi$ be the set of system performance features that should be limited in variation. For each element $\phi_i \in \Phi$, quantitatively describe the tolerable variation in $\phi_i$. Let $\left\langle \beta_i^{\min}, \beta_i^{\max} \right\rangle$ be a tuple that gives the bounds of the tolerable variation in the system feature $\phi_i$.

2) Identify all of the system and environment parameters whose values may impact the QoS performance features selected in step 1. These are called the *perturbation parameters*, and the performance features are required to be robust with respect to these perturbation parameters. Mathematically, let $\Pi$ be the set of perturbation parameters. It is assumed that the elements of $\Pi$ are vectors. Let $\boldsymbol{\pi}_j$ be the $j$-th element of $\Pi$. For the makespan example, $\boldsymbol{\pi}_j$ could be the vector composed of the actual application execution times, i.e., the $i$-th element of $\boldsymbol{\pi}_j$ is the actual execution time of the $i$-th application on the machine it was assigned. In general, representation of the perturbation parameters as separate elements of $\Pi$ would be based on their nature or kind (e.g., message length variables in $\boldsymbol{\pi}_1$ and computation time variables in $\boldsymbol{\pi}_2$).

3) Identify the impact of the perturbation parameters in step 2 on the system performance features in step 1. Mathematically, for every $\phi_i \in \Phi$, determine the relationship $\phi_i = f_{ij}(\boldsymbol{\pi}_j)$, if any, that relates $\phi_i$ to $\boldsymbol{\pi}_j$. In this expression, $f_{ij}$ is a function that maps $\boldsymbol{\pi}_j$ to $\phi_i$.

4) The last step is to determine the smallest collective variation in the values of perturbation parameters identified in step 2 that will cause any of the performance features identified in step 1 to violate its acceptable variation. This will be the degree of robustness of the given resource allocation.

Mathematically, for every $\phi_i \in \Phi$, determine the *boundary values of* $\boldsymbol{\pi}_j$*,* i.e., the values satisfying the *boundary relationships* $f_{ij}(\boldsymbol{\pi}_j) = \beta_i^{\min}$ and $f_{ij}(\boldsymbol{\pi}_j) = \beta_i^{\max}$. These relationships separate the region of robust operation from that of non-robust operation. Find the smallest perturbation in $\boldsymbol{\pi}_j$ that causes any $\phi_i \in \Phi$ to exceed the bounds $\left\langle \beta_i^{\min}, \beta_i^{\max} \right\rangle$ imposed on it by the robustness requirement.

Figure 1 illustrates this concept for a single feature, $\phi_i$, and a two-element perturbation vector $\boldsymbol{\pi}_j \in \mathbf{R}^2$. The curve shown in Figure 1 plots the set of boundary points $\{\boldsymbol{\pi}_j |\ f_{ij}(\boldsymbol{\pi}_j) = \beta_i^{\max}\}$ for a resource allocation $\mu$. For this figure, the set of boundary points $\left\{\boldsymbol{\pi}_j |\ f_{ij}(\boldsymbol{\pi}_j) = \beta_i^{\min}\right\}$ is given by the points on the $\pi_{j1}$-axis and $\pi_{j2}$-axis.

The region enclosed by the axes and the curve gives the values of $\boldsymbol{\pi}_j$ for which the system is robust with respect to $\phi_i$. For a vector $\mathbf{x} = [x_1\, x_2\, \cdots\, x_n]^{\mathrm{T}}$, let $\|\mathbf{x}\|_2$ be the $\ell_2$-norm (Euclidean norm) of the vector, defined by $\sqrt{\sum_{r=1}^{n} x_r^2}$. The point on the curve marked as $\boldsymbol{\pi}_j^{\star}(\phi_i)$ has the property that the Euclidean distance from $\boldsymbol{\pi}_j^{\text{orig}}$ to $\boldsymbol{\pi}_j^{\star}(\phi_i)$, $\|\boldsymbol{\pi}_j^{\star}(\phi_i) - \boldsymbol{\pi}_j^{\text{orig}}\|_2$, is the smallest over all such distances from $\boldsymbol{\pi}_j^{\text{orig}}$ to a point on the curve, and is defined to be the *robustness radius*, $r_\mu(\phi_i,\ \boldsymbol{\pi}_j)$, of $\phi_i$ against $\boldsymbol{\pi}_j$. Mathematically,

$$r_\mu(\phi_i,\ \boldsymbol{\pi}_j) = \min_{\boldsymbol{\pi}_j:\,(f_{ij}(\boldsymbol{\pi}_j)=\beta_i^{\max})\vee(f_{ij}(\boldsymbol{\pi}_j)=\beta_i^{\min})} \|\boldsymbol{\pi}_j - \boldsymbol{\pi}_j^{\text{orig}}\|_2. \quad (1)$$

The quantity $\rho_\mu(\Phi,\ \boldsymbol{\pi}_j)$ is defined as the robustness of resource allocation $\mu$ with respect to the performance feature set $\Phi$ against the perturbation parameter $\boldsymbol{\pi}_j$, and is given by the minimum of all robustness radii, or $\rho_\mu(\Phi,\ \boldsymbol{\pi}_j) = \min_{\phi_i \in \Phi}\left(r_\mu(\phi_i,\ \boldsymbol{\pi}_j)\right)$.

## 3. Robustness Against Multiple Perturbation Parameters

### 3.1. Overview

The research in [2] developed the analysis for determining the robustness metric for a system with a single perturbation parameter. In this section, that analysis is extended to include multiple perturbation parameters.

Consider a HiPer-D like system where task execution times, $e_j$, and message lengths $m_k$, change unpredictably. Also assume that the performance feature, $\phi_i$, is a function of $e_j$ and $m_k$. Given such a function for $\phi_i$, one could use the method summarized in Section 2 to "determine" $\rho_\mu(\Phi,\ \boldsymbol{\pi}_j)$. However, the method in Section 2 assumes that all elements of $\boldsymbol{\pi}_j$ have the same units. Because $e_j$ and $m_k$ have different units, one cannot assemble all of them in one perturbation parameter, $\boldsymbol{\pi}_j$, and then determine $\rho_\mu(\Phi,\ \boldsymbol{\pi}_j)$.

Furthermore, [2] makes it clear that the unit of $\rho_\mu(\Phi,\ \boldsymbol{\pi}_j)$ is the same as that for any element of $\boldsymbol{\pi}_j$. Given this fact, any calculation for $\rho_\mu(\Phi,\ \boldsymbol{\pi}_j)$ would be questionable even if one did assemble all $e_j$ and $m_k$ in one perturbation parameter. This further clarifies why one cannot assemble $e_j$ and $m_k$ in one $\boldsymbol{\pi}_j$ without first adjusting for the unit changes.

One can argue that $\phi_i$ includes necessary conversions of multiple perturbation parameters which are not of the same kind. However, these conversions are not included in the robustness measurement equation. That is the reason why we have to merge $\boldsymbol{\pi}_j$'s into **P** so as to compose a dimensionless perturbation parameter vector.

Multiple perturbation parameters are considered in [2] by concatenating them into one parameter, which is then used as a single parameter as discussed in Section 2. Then the robustness metric is determined by taking the minimum over the robustness radii of all $\phi_i \in \Phi$.

Let the vector $\boldsymbol{\pi}_j$ have $\underline{n_{\boldsymbol{\pi}_j}}$ elements, and let $\star$ be the vector concatenation operator, so that $\boldsymbol{\pi}_1 \star \boldsymbol{\pi}_2 = [\ \pi_{11}\ \ \pi_{12}\ \cdots\ \pi_{1n_{\pi_1}}\ \ \pi_{21}\ \ \pi_{22}\ \cdots\ \pi_{2n_{\pi_2}}\ ]^{\mathrm{T}}$. Let $\underline{\mathbf{P}} \in \mathbb{P}$ be a weighted concatenation of the vectors $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_{|\Pi|}$, where $\mathbb{P}$ is a space of $n_{\pi_1} + n_{\pi_2} + \cdots + n_{\pi_{|\Pi|}}$ dimensions. That is, $\mathbf{P} = (\alpha_1 \times \boldsymbol{\pi}_1) \star (\alpha_2 \times \boldsymbol{\pi}_2) \star \cdots \star (\alpha_{|\Pi|} \times \boldsymbol{\pi}_{|\Pi|})$, where $\underline{\alpha_j}$ $(1 \leq j \leq |\Pi|)$ is a weighting constant.

The vector **P** is analogous to the vector $\boldsymbol{\pi}_j$ discussed in Section 2. Parallel to the discussion in Section 2, one needs to identify the set of boundary values of **P**. Let $\underline{f_i}$ be a function that maps **P** to $\phi_i$. For the single system feature $\phi_i$ being considered, such a set is given by $\left\{\mathbf{P} \,\middle|\, (f_i(\mathbf{P}) = \beta_i^{\max}) \bigvee (f_i(\mathbf{P}) = \beta_i^{\min})\right\}$.

Let $\underline{\mathbf{P}^{\text{orig}}}$ be the assumed value of **P**. In addition, let $\underline{\mathbf{P}^{\star}(\phi_i)}$ be, analogous to $\boldsymbol{\pi}_j^{\star}(\phi_i)$, the element in the set of boundary values such that the Euclidean distance from $\mathbf{P}^{\text{orig}}$ to $\mathbf{P}^{\star}(\phi_i)$, $\|\mathbf{P}^{\star}(\phi_i) - \mathbf{P}^{\text{orig}}\|_2$, is the smallest over all such distances from $\mathbf{P}^{\text{orig}}$ to a point in the boundary set. Alternatively, the value $\|\mathbf{P}^{\star}(\phi_i) - \mathbf{P}^{\text{orig}}\|_2$ gives the largest Euclidean distance that the variable **P** can move in *any* direction from an assumed value of $\mathbf{P}^{\text{orig}}$ without exceeding the tolerable limits on $\phi_i$. Parallel to the discussion in Section 2, let the distance $\|\mathbf{P}^{\star}(\phi_i) - \mathbf{P}^{\text{orig}}\|_2$ be called the robustness radius, $r_\mu(\phi_i,\ \mathbf{P})$, of $\phi_i$ against **P**. Mathematically,

$$r_\mu(\phi_i,\ \mathbf{P}) = \min_{\mathbf{P}:\,(f_i(\mathbf{P})=\beta_i^{\max}) \bigvee (f_i(\mathbf{P})=\beta_i^{\min})} \|\mathbf{P} - \mathbf{P}^{\text{orig}}\|_2. \tag{2}$$

Extending for all $\phi_i \in \Phi$, the robustness of resource allocation $\mu$ with respect to the performance feature set $\Phi$ against the perturbation parameter set $\Pi$ is given by $\rho_\mu(\Phi,\ \mathbf{P}) = \min_{\phi_i \in \Phi} \left(r_\mu(\phi_i,\ \mathbf{P})\right)$.

So how can one use the value of robustness calculated in $\mathbb{P}$-space? How can one relate this value to allowable changes in $\boldsymbol{\pi}_j$'s? Consider a system that has perturbation parameters of different kinds. Assume that its robustness value, as calculated in $\mathbb{P}$-space, is $r_\mu(\phi_i, \mathbf{P})$ for a set of $\boldsymbol{\pi}_j^{\text{orig}}$ values. To find out whether the system can operate without a constraint violation under a given set of $\boldsymbol{\pi}_j$ values, one can (a) convert the $\boldsymbol{\pi}_j$ values into a **P** value using the $\alpha_j$'s, (b) calculate the distance of **P** from $\mathbf{P}^{\text{orig}}$ as $\|\mathbf{P} - \mathbf{P}^{\text{orig}}\|_2$, and (c) determine if $\|\mathbf{P} - \mathbf{P}^{\text{orig}}\|_2 < r_\mu(\phi_i, \mathbf{P})$. If yes, then the system will not violate a constraint when operated at the given values of $\boldsymbol{\pi}_j$'s.

The sensitivity-based weighting procedure for the calculation of $\alpha_j$'s is now discussed. Typically, $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, $\cdots$, $\boldsymbol{\pi}_{|\Pi|}$ will have different dimensions, i.e., will be measured in different units, e.g., seconds, objects per data set, bytes, etc. Before the concatenation of these vectors into **P**, they should be converted into a single dimension. The proposed preliminary approach in [2] suggested a sensitivity-based weighting, that is: $\alpha_j = 1/r_\mu(\phi_i,\ \boldsymbol{\pi}_j)$. With this definition of $\alpha_j$,

$$\mathbf{P} = \frac{\boldsymbol{\pi}_1}{r_\mu(\phi_i,\ \boldsymbol{\pi}_1)} \star \frac{\boldsymbol{\pi}_2}{r_\mu(\phi_i,\ \boldsymbol{\pi}_2)} \star \cdots \star \frac{\boldsymbol{\pi}_{|\Pi|}}{r_\mu(\phi_i,\ \boldsymbol{\pi}_{|\Pi|})}.$$

Note that the units of $r_\mu(\phi_i,\ \boldsymbol{\pi}_j)$ are the units of $\boldsymbol{\pi}_j$. This fact renders **P** dimensionless. Therefore the robustness measurement, $r_\mu(\phi_i, \mathbf{P})$, would be dimensionless as well. However, the investigation done in this research shows that the sensitivity-based weighting method has the following problem. When the performance feature is a linear function of one-element perturbation parameters, the robustness radius depends <u>only</u> on the number of perturbation parameters. Any change in the coefficients of $\phi_i$ function, or the original values of the perturbation parameters does not affect the robustness radius. This means that if two systems have performance features that are linear functions of the same number of perturbation parameters, then they will have the same robustness radius. With such characteristic in a robustness measure, it is impossible to compare the robustness of different systems that meet the criterion given above.

The above point is now illustrated in more detail. Let $\phi_i(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_n)$ be a function of *n* perturbation vectors, $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_n$, of different kinds. Assume that vectors $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_n$ have only one element each, i.e., $\boldsymbol{\pi}_1 = \pi_{11} = \pi_1$, $\boldsymbol{\pi}_2 = \pi_{21} = \pi_2, \cdots, \boldsymbol{\pi}_n = \pi_{n1} = \pi_n$. The discussion below will consider only $\beta_i^{\max}$ as a constraint.

As a general linear case, let $\phi_i(\pi_1, \pi_2, \cdots, \pi_n) = k_1\pi_1 + k_2\pi_2 + \cdots + k_n\pi_n$, and the original values of the perturbation parameters be $\pi_1^{\text{orig}}, \pi_2^{\text{orig}}, \cdots, \pi_n^{\text{orig}}$, respectively, and $\beta_i^{\max} = \beta\phi_i^{\text{orig}} = \beta(k_1\pi_1^{\text{orig}} + k_2\pi_2^{\text{orig}} + \cdots + k_n\pi_n^{\text{orig}})$, where $\beta$ is an arbitrary constant greater than 1. Note that this implies that $\beta_i^{\max}$ is a function of original value of the performance feature, $\phi_i$. This assumption does make sense because in many cases we limit the changes in $\phi_i$ to some percentage of its original value (e.g. in a Grid-like system, makespan should not exceed 1.2 times its original value). The following steps are performed to calculate the robustness radius.

**Step 1**: Determine the robustness radius with respect to $\pi_j$, $r_\mu(\phi_i, \pi_j)$ by setting $\pi_m$, $m \neq j$, to $\pi_m^{\text{orig}}$ in the $\phi_i$ function. Then let $\alpha_j$ be $1/r_\mu(\phi_i, \pi_j)$:

Example for $\pi_1$

$\pi_m = \pi_m^{\text{orig}}, \;\; m \neq 1$:

The constraint equation is $\phi_i|_{\substack{\pi_m = \pi_m^{\text{orig}} \\ m \neq 1}} = \beta_i^{\max}$, or

$$k_1\pi_1 + k_2\pi_2^{\text{orig}} + \cdots + k_n\pi_n^{\text{orig}} = \beta(k_1\pi_1^{\text{orig}} + k_2\pi_2^{\text{orig}} + \cdots + k_n\pi_n^{\text{orig}}).$$

Solving the above equation for $\pi_1$,

$$\pi_1 = \frac{\beta k_1\pi_1^{\text{orig}} + (\beta - 1)(k_2\pi_2^{\text{orig}} + \cdots + k_n\pi_n^{\text{orig}})}{k_1}.$$

Now the robustness radius can be calculated by applying Equation 1. Since it is a one-dimensional space, the minimum Euclidean distance between $\pi_1^{\text{orig}}$ and the constraint curve is just the distance between $\pi_1^{\text{orig}}$ and the point $\pi_1$ which was calculated above. That is $r_\mu(\phi_i, \pi_1) = \pi_1 - \pi_1^{\text{orig}}$, therefore

$$r_\mu(\phi_i, \pi_1) = \frac{\beta - 1}{k_1}(k_1\pi_1^{\text{orig}} + k_2\pi_2^{\text{orig}} + \cdots + k_n\pi_n^{\text{orig}})$$

$$\begin{aligned} \alpha_1 &= 1/r_\mu(\phi_i, \pi_1) \\ &= \frac{k_1}{(\beta - 1)(k_1\pi_1^{\text{orig}} + k_2\pi_2^{\text{orig}} + \cdots + k_n\pi_n^{\text{orig}})}. \end{aligned} \quad (3)$$

For the general case, $\pi_m = \pi_m^{\text{orig}}, \;\; m \neq j$:

The constraint equation is $\phi_i|_{\substack{\pi_m = \pi_m^{\text{orig}} \\ m \neq j}} = \beta_i^{\max}$, or

$$\sum_{\substack{m=1 \\ m \neq j}}^{n} k_m\pi_m^{\text{orig}} + k_j\pi_j = \beta \sum_{m=1}^{n} k_m\pi_m^{\text{orig}}.$$

Solving the above equation for $\pi_j$,

$$\pi_j = \frac{\beta k_j\pi_j^{\text{orig}} + (\beta - 1)\displaystyle\sum_{m=1,\, m \neq j}^{n} k_m\pi_m^{\text{orig}}}{k_j}$$

As before, since it is a one-dimensional space, the minimum Euclidean distance between $\pi_j^{\text{orig}}$ and the constraint curve is just the distance between $\pi_j^{\text{orig}}$ and the point $\pi_j$ which was calculated above. That is $r_\mu(\phi_i, \pi_j) = \pi_j - \pi_j^{\text{orig}}$, then

$$r_\mu(\phi_i, \pi_j) = \frac{\beta - 1}{k_j}\sum_{m=1}^{n} k_m\pi_m^{\text{orig}}$$

$$\begin{aligned} \alpha_j &= 1/r_\mu(\phi_i, \pi_j) \\ &= \frac{k_j}{(\beta - 1)\displaystyle\sum_{m=1}^{n} k_m\pi_m^{\text{orig}}}. \end{aligned}$$

**Step 2**: Using the $\alpha_j$'s obtained in Step 1, determine the relationships between corresponding elements of **P** and $\pi_j$'s. Deduce the function $f_i$ as a function of **P** from these relationships, and then generate a robustness metric in $\mathbb{P}$-space.

$\mathbf{P} = [\alpha_1\pi_1 \;\; \alpha_2\pi_2 \;\; \cdots \;\; \alpha_n\pi_n]$.

The constraint equation is $\phi_i = \beta_i^{\max}$, where the left-hand side would be simplified by replacing $\pi_j$'s with their equivalents in $P_j$'s,

$$\begin{aligned} \phi_i &= k_1\pi_1 + k_2\pi_2 + \cdots + k_n\pi_n \\ &= k_1P_1/\alpha_1 + k_2P_2/\alpha_2 + \cdots + k_nP_n/\alpha_n. \end{aligned}$$

Given that

$$\alpha_j = \frac{k_j}{(\beta - 1)\displaystyle\sum_{m=1}^{n} k_m\pi_m^{\text{orig}}},$$

the $\phi_i$ equation would be simplified as:

$$\phi_i = k_1 P_1 \frac{(\beta-1)\sum_{m=1}^{n} k_m \pi_m^{\text{orig}}}{k_1} + \cdots$$
$$+ k_n P_n \frac{(\beta-1)\sum_{m=1}^{n} k_m \pi_m^{\text{orig}}}{k_n}$$
$$= (P_1 + P_2 + \cdots + P_n)(\beta-1)\sum_{m=1}^{n} k_m \pi_m^{\text{orig}}.$$

The right-hand side of the constraint equation, $\beta_i^{\max}$, would be:

$$\begin{aligned}\beta_i^{\max} &= \beta\phi_i^{\text{orig}} \\ &= \beta(k_1\pi_1^{\text{orig}} + k_2\pi_2^{\text{orig}} + \cdots + k_n\pi_n^{\text{orig}}) \\ &= \beta\sum_{m=1}^{n} k_m \pi_m^{\text{orig}}.\end{aligned}$$

Therefore the constraint equation, $\phi_i = \beta_i^{\max}$, would be simplified as:
$P_1 + P_2 + \cdots + P_n = \beta/(\beta-1)$.

Then the robustness radius of $\phi_i$ with respect to $\mathbf{P}$, that is the minimum Euclidean distance from $\mathbf{P}^{\text{orig}}$ to the $n$-dimensional plane above. Recall that given an $n$-dimensional plane, $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$, the minimum distance between a point $X_0 : (x_{01}, x_{02}, \cdots, x_{0n})$ and the plane is would be:

$$d = \frac{|a_1x_{01} + a_2x_{02} + \cdots + a_nx_{0n} - b|}{\sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}}. \qquad (4)$$

Then $r_\mu(\phi_i, \mathbf{P})$ would be:

$$r_\mu(\phi_i, \mathbf{P}) = \frac{|P_1^{\text{orig}} + P_2^{\text{orig}} + \cdots + P_n^{\text{orig}} - \beta/(\beta-1)|}{\sqrt{n}}.$$

The above equation can be furthur simplified by replacing $P_j^{\text{orig}}$'s by their equivalents, $\alpha_j\pi_j^{\text{orig}}$, where

$$\alpha_j\pi_j^{\text{orig}} = \frac{k_j\pi_j^{\text{orig}}}{(\beta-1)\sum_{m=1}^{n} k_m\pi_m^{\text{orig}}},$$

We will get:

$$\begin{aligned} r_\mu(\phi_i, \mathbf{P}) &= \frac{\left|\frac{\sum_{j=1}^{n} k_j\pi_j^{\text{orig}}}{(\beta-1)\sum_{m=1}^{n} k_m\pi_m^{\text{orig}}} - \frac{\beta}{\beta-1}\right|}{\sqrt{n}} \\ &= \frac{\left|\frac{1}{\beta-1} - \frac{\beta}{\beta-1}\right|}{\sqrt{n}} \\ &= \frac{1}{\sqrt{n}}.\end{aligned}$$

One can see that regardless of the values of $k_j$'s, $\beta$ and the original values of $\pi_j$'s, the robustness radius is equal to $\frac{1}{\sqrt{n}}$. The fact that an increase in the robustness requirement, $\beta_i^{\max}$, does not change the robustness value is troubling. This may not be a desired characteristic of a robustness measure. Therefore the sensitivity-based concatenation may not give a satisfactory robustness measure.

### 3.2. Normalizing the Robustness Measure with respect to the Original Values of Perturbation Parameters

The purpose of this section is to propose a method for robustness measure calculation which can be used for multiple kinds of perturbations, and be able to compare different cases. We still have to merge $\boldsymbol{\pi}_j$'s into $\mathbf{P}$ so as to compose a dimensionless perturbation parameter vector, and then determine the robustness metric by taking the minimum over the robustness radii of all $\phi_i \in \Phi$, as described in the Overview section.

Let the vector $\boldsymbol{\pi}_j$ have $n_{\boldsymbol{\pi}_j}$ elements $(1 \leq j \leq |\Pi|)$, that is $\boldsymbol{\pi}_j = [\pi_{j1} \quad \pi_{j2} \quad \cdots \quad \pi_{jn_{\boldsymbol{\pi}_j}}]$. Also redefine $\mathbf{P}$ as a concatenation of vectors $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_{|\Pi|}$ normalized by the original values of their elements:

$$\begin{aligned}\mathbf{P} =& [\pi_{11}/\pi_{11}^{\text{orig}} \quad \pi_{12}/\pi_{12}^{\text{orig}} \quad \cdots \quad \pi_{1n_{\pi_1}}/\pi_{1n_{\pi_1}}^{\text{orig}}] \star \\ & \cdots \star \\ & [\pi_{|\Pi|1}/\pi_{|\Pi|1}^{\text{orig}} \quad \pi_{|\Pi|2}/\pi_{|\Pi|2}^{\text{orig}} \quad \cdots \quad \pi_{|\Pi|n_{\pi_{|\Pi|}}}/\pi_{|\Pi|n_{\pi_{|\Pi|}}}^{\text{orig}}].\end{aligned} \qquad (5)$$

Now $\mathbf{P}^{\text{orig}}$ will always be $[1 \quad 1 \cdots 1]$. Using this definition, the elements of $\mathbf{P}$ would be dimensionless, therefore $r_\mu(\phi_i, \mathbf{P})$ would be dimensionless as well. The quantity $\rho_\mu(\Phi,\ \mathbf{P})$ is defined as the robustness of resource allocation $\mu$ with respect to the performance feature set $\Phi$ against the perturbation parameter set $\Pi$, and is given by the minimum of all robustness radii, or $\rho_\mu(\Phi,\ \mathbf{P}) = \min_{\phi_i \in \Phi} \left(r_\mu(\phi_i,\ \mathbf{P})\right)$.

Furthermore, in this definition, the relative, not the absolute, changes of each perturbation parameter element with respect to its original value are examined. We believe that this method can be applied on systems with a single perturbation parameter as well as the systems with multiple perturbation parameters, to achieve a uniform dimensionless robustness measure. And this method does not have the problem present in [2] which was discussed in Section 3.1.

For the same general linear case which was discussed in Section 3.1, $r_\mu(\phi_i, \mathbf{P})$ is calculated as following. The concatenated vector, **P**, is derived from Equation (4):
$\mathbf{P} = [\pi_1/\pi_1^{\text{orig}} \ \ \pi_2/\pi_2^{\text{orig}} \ \ \cdots \ \ \pi_n/\pi_n^{\text{orig}}]$.
The left-hand side of the constraint equation, $\phi_i = \beta_i^{\max}$, would be simplified by replacing $\pi_j$'s with their equivalents in $P_j$'s,

$$\begin{aligned}\phi_i &= k_1\pi_1 + k_2\pi_2 + \cdots + k_n\pi_n \\ &= k_1\pi_1^{\text{orig}}P_1 + k_2\pi_2^{\text{orig}}P_2 + \cdots + k_n\pi_n^{\text{orig}}P_n.\end{aligned}$$

The right-hand side of the constraint equation, $\beta_i^{\max}$, would be:

$$\begin{aligned}\beta_i^{\max} &= \beta\phi_i^{\text{orig}} \\ &= \beta(k_1\pi_1^{\text{orig}} + k_2\pi_2^{\text{orig}} + \cdots + k_n\pi_n^{\text{orig}}) \\ &= \beta\sum_{m=1}^{n} k_m\pi_m^{\text{orig}}.\end{aligned}$$

Therefore the constraint equation, $\phi_i = \beta_i^{\max}$, would be simplified as:

$$\sum_{j=1}^{n} k_j\pi_j^{\text{orig}}P_j = \beta\sum_{m=1}^{n} k_m\pi_m^{\text{orig}}$$

$$k_1\pi_1^{\text{orig}}P_1 + k_2\pi_2^{\text{orig}}P_2 + \cdots + k_n\pi_n^{\text{orig}}P_n = \beta\sum_{m=1}^{n} k_m\pi_m^{\text{orig}}.$$

Then the robustness radius of $\phi_i$ with respect to **P**, that is the minimum Euclidean distance from $\mathbf{P}^{\text{orig}}$ to the $n$-dimensional plane above. Using Equation 4, the robustness radius would be:

$$r_\mu(\phi_i, \mathbf{P}) = \frac{\left|\sum_{j=1}^{n} k_j\pi_j^{\text{orig}}P_j^{\text{orig}} - \beta\sum_{m=1}^{n} k_m\pi_m^{\text{orig}}\right|}{\sqrt{\sum_{m=1}^{n}(k_m\pi_m^{\text{orig}})^2}}.$$

Recalling that $\mathbf{P} = [1 \quad 1\cdots 1]$, the above equation is simplified to:

$$\begin{aligned}r_\mu(\phi_i, \mathbf{P}) &= \frac{\left|\sum_{j=1}^{n} k_j\pi_j^{\text{orig}} - \beta\sum_{m=1}^{n} k_m\pi_m^{\text{orig}}\right|}{\sqrt{\sum_{m=1}^{n}(k_m\pi_m^{\text{orig}})^2}} \\ &= (\beta - 1)\frac{\left|\sum_{j=1}^{n} k_j\pi_j^{\text{orig}}\right|}{\sqrt{\sum_{m=1}^{n}(k_m\pi_m^{\text{orig}})^2}}.\end{aligned}$$

One can see that now the robustness radius depends, as it should, on the values of $k_j$'s, $\beta$, and the original values of $\pi_j$'s.

## 4. Conclusions

Parallel and distributed heterogeneous computing systems may operate in environments that undergo unpredicted changes in environment or component parts of system. A measurement method for the robustness of such systems has been presented and developed in [2]. However, that approach was mainly focused on systems in which all perturbation parameters were of the same kind. In our paper, we focus on the multiple perturbation parameter case and on merging the different perturbation parameters into a single perturbation parameter. The investigation done in this research further strengthens the sensitivity-based weighting method presented in [2]. When the performance feature is a linear function of one-element perturbation parameters, the robustness radius given in [2] depends only on the number of perturbation parameters. In other words, regardless of the values of the robustness requirement, the coefficients of $\phi_i$ function, or the original values of the perturbation parameters, the robustness radius is equal to $\frac{1}{\sqrt{n}}$, where $n$ is the number of perturbation parameters. This means that if two systems have performance features that are linear functions of the same number of perturbation parameters, then they will have the same robustness radius. We propose an alternative robustness metric formulation that avoids the problem mentioned above.

## References


[1] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," *IEEE International Conference on Network Protocols (ICNP'01)*, Nov. 2001.

[2] S. Ali, A. A. Maciejewski, H. J. Siegel, and J.-K. Kim, "Measuring the robustness of a resource allocation," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 7, July 2004, pp. 630–641.

[3] L. Bölöni and D. C. Marinescu, "Robust scheduling of metaprograms," *Journal of Scheduling*, Vol. 5, No. 5, Sept. 2002, pp. 395–412.

[4] J. M. Carlson and J. Doyle, "Complexity and robustness," *Proceedings of National Academy of Science (PNAS)*, Vol. 99, No. 1, Feb. 2002, pp. 2538–2545.

[5] A. J. Davenport, C. Gefflot, and J. C. Beck, "Slack-based techniques for robust schedules," *6th European Conference on Planning (ECP-2001)*, Sept. 2001, pp. 7–18.

[6] J. DeVale and P. Koopman, "Robust software – no more excuses," *IEEE International Conference on Dependable Systems and Networks (DSN '02)*, June 2002, pp. 145–154.

[7] J. DeVale, *High Performance Robust Computer Systems*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 2001.

[8] E. Hart, P. M. Ross, and J. Nelson, "Producing robust schedules via an artificial immune system," *1998 International Conference on Evolutionary Computing*, May 1998, pp. 464–469.

[9] D. E. Hastings, A. L. Weigel, and M. A. Walton, "Incorporating uncertainty into conceptual design of space system architectures," *ESD Internal Symposium, Working Paper Series*, May 2002, ESD-WP-2003-01.01.

[10] IRIS: Infrastructure for Resilient Internet Systems, "IRIS ITR proposal," `http://project-iris.net/proposal.html`, 2001.

[11] M. Jensen, "Improving robustness and flexibility of tardiness and total flowtime job shops using robustness measures," *Journal of Applied Soft Computing*, Vol. 1, No. 1, June 2001, pp. 35–52.

[12] V. J. Leon, S. D. Wu, and R. H. Storer, "Robustness measures and robust scheduling for job shops," *IEE Transactions*, Vol. 26, No. 5, Sept. 1994, pp. 32–43.

[13] R. Rodrigues, B. Liskov, and L. Shrira, "The design of a robust peer-to-peer system," *10th ACM SIGOPS European Workshop*, Sept. 2002, pp. 22–25.

[14] M. Sevaux and K. Sörensen, "Genetic algorithm for robust schedules," *8th International Workshop on Project Management and Scheduling (PMS 2002)*, Apr. 2002, pp. 330–333.

[15] F. Ye, G. Zhong, S. Lu, and L. Zhang, "A robust data delivery protocol for large scale sensor networks," *2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*, Apr. 2003.

## Biographies

**Behdis Eslamnour** received the MS degree in electrical engineering in 2000 from the Department of Electrical Engineering at the Amirkabir University of Technology, Tehran, Iran. She received the BS degree in electrical engineering in 1996 from the Sharif University of Technology, Tehran, Iran. She is a graduate student in the Department of Electrical and Computer Engineering at the University of Missouri-Rolla, where she is currently a research assistant. Her research interests include heterogeneous distributed computing and resource management, and sensor networks. She is a student member of IEEE.

**Shoukat Ali** received the MSEE (1999) and PhD (2003) degrees from the School of Electrical and Computer Engineering at Purdue University, West Lafayette. He received the BS degree (1996) in electrical engineering from the University of Engineering and Technology (UET), Lahore, Pakistan. He is an assistant professor in the Department of Electrical and Computer Engineering at the University of Missouri-Rolla. His research interests include heterogeneous parallel and distributed computing and communication systems. He has coauthored 18 published technical papers in this area. He was the publicity co-chair of the 11th IEEE Heterogeneous Computing Workshop (2002), the program committee member for the 12th and 14th IEEE Heterogeneous Computing Workshops, and the proceedings chair and program committee member for IPDPS 2005. He is a member of the IEEE and IEEE Computer Society.

| | |
|---|---|
| $\Phi$ | the set of all performance features |
| $\phi_i$ | the $i$-th element in $\Phi$ |
| $\langle \beta_i^{\min}, \beta_i^{\max} \rangle$ | a tuple that gives the bounds of the tolerable variation in $\phi_i$ |
| $\Pi$ | the set of all perturbation parameters |
| $\boldsymbol{\pi}_j$ | the $j$-th element in $\Pi$ |
| $f_{ij}$ | the function that maps $\boldsymbol{\pi}_j$ to $\phi_i$ |
| $n_{\boldsymbol{\pi}_j}$ | the dimension of vector $\boldsymbol{\pi}_j$ |
| $\mu$ | a resource allocation |
| $r_\mu(\phi_i,\ \boldsymbol{\pi}_j)$ | the robustness radius of resource allocation $\mu$ with respect to $\phi_i$ against $\boldsymbol{\pi}_j$ |
| $\rho_\mu(\Phi,\ \boldsymbol{\pi}_j)$ | the robustness of resource allocation $\mu$ with respect to set $\Phi$ against $\boldsymbol{\pi}_j$ |
| $f_i$ | the function that maps $\mathbf{P}$ to $\phi_i$ |
| $\mathbf{P}$ | a weighted concatenation of the vectors $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_{\|\Pi\|}$ |
| $r_\mu(\phi_i,\ \mathbf{P})$ | the robustness radius of resource allocation $\mu$ with respect to $\phi_i$ against $\mathbf{P}$ |
| $\rho_\mu(\Phi,\ \mathbf{P})$ | the robustness of resource allocation $\mu$ with respect to set $\Phi$ against $\mathbf{P}$ |

**Table 1. Glossary of notation.**

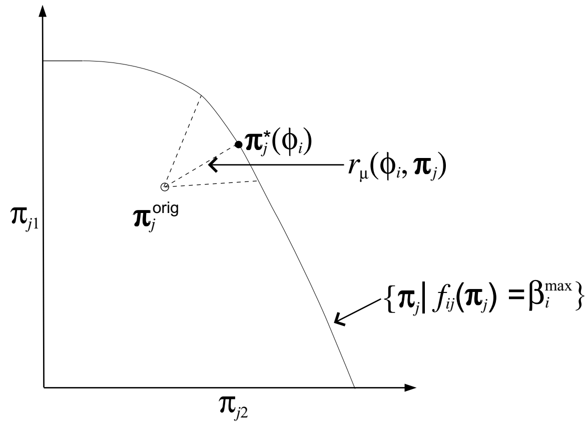


**Figure 1. Some possible directions of increase of the perturbation parameter $\pi_j$, and the direction of the smallest increase. The curve plots the set of points, $\{\boldsymbol{\pi}_j |\ f_{ij}(\boldsymbol{\pi}_j) = \beta_i^{\max}\}$. The set of boundary points, $\{\boldsymbol{\pi}_j |\ f_{ij}(\boldsymbol{\pi}_j) = \beta_i^{\min}\}$ is given by the points on the $\pi_{j1}$-axis and $\pi_{j2}$-axis.**

Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)
1530-2075/05 $ 20.00 **IEEE**