Masi, Luca and Vasile, Massimiliano (2012) *Optimal multi-objective discrete decision making using a multidirectional modified Physarum solver.* In: EVOLVE 2012 International Conference, 2012-08-07 - 2012-08-09, Mexico City.

OPTIMAL MULTI-OBJECTIVE DISCRETE DECISION MAKING USING A MULTIDIRECTIONAL MODIFIED *PHYSARUM* SOLVER

L. Masi[a] and **M. Vasile**[b]

[a]Department of Mechanical & Aerospace Engineering
University of Strathclyde
75 Montrose Street, G1 1XJ, Glasgow, UK
luca.masi@strath.ac.uk

[b]Department of Mechanical & Aerospace Engineering
University of Strathclyde
75 Montrose Street, G1 1XJ, Glasgow, UK
massimiliano.vasile@strath.ac.uk

Abstract

This paper will address a bio-inspired algorithm able to incrementally grow decision graphs in multiple directions for discrete multi-objective optimization. The algorithm takes inspiration from the slime mould *Physarum Polycephalum*, an amoeboid organism that in its plasmodium state extends and optimizes a net of veins looking for food. The algorithm is here used to solve multi-objective Traveling Salesman and Vehicle Routing Problems selected as representative examples of multi-objective discrete decision making problems. Simulations on selected test case showed that building decision sequences in two directions and adding a matching ability (multidirectional approach) is an advantageous choice if compared with the choice of building decision sequences in only one direction (unidirectional approach). The ability to evaluate decisions from multiple directions enhances the performance of the solver in the construction and selection of optimal decision sequences.

1 Introduction

Over the past two decades, bio-inspired computation has become an appealing topic to solve NP-hard problems in combinatorial optimization. The method proposed in this paper takes inspiration from *Physarum Polycephalum*, a simple amoeboid organism that was endowed by nature with heuristics that can be used to solve multi-objective discrete decision making problems. In [9] it has been shown that *Physarum Polycephalum* is able to solve a maze finding the shortest path that connects the maze's entrance and exit by changing its shape. It has been shown also that a living *Physarum* is able to recreate the Japan rail network [12] and the Mexican highway network [1], both using an experimental arena with food sources at each of the major cities in the regions. *Physarum* based algorithms have been developed recently to solve multi-source problems with a simple geometry [5, 10], mazes [11] and transport network problems [12, 11]. In this paper a multidirectional modified *Physarum Polycephalum* algorithm able to solve NP-hard multi-objective classical problems in operations research is proposed. In [2, 6] multi-objective bio-inspired algorithms, i.e. ant colony algorithms, have been proposed and studied. The algorithm is a multi-objective generalization of the single-objective multidirectional modified *Physarum* solver previously presented in [7] for discrete decision making. In Sect. 2 the physiology of *Physarum* is introduced: discrete decision making problems are modeled with decision graphs where nodes represent the possible decisions while arcs represent the cost vector associated with decisions. Each arc has a scalar dominance index associated. Decision graphs are incrementally grown and explored in multiple directions using the *Physarum*-based heuristic.

This paper aims at proving that a multidirectional incremental *Physarum* solver is more efficient, in terms of success indexes (see Sect. 3.1), than a unidirectional incremental *Physarum* solver when applied to the solution of multi-objective decision problems that can be represented with directed symmetric decision graphs, i.e. graphs where the contribution of an arc to a complete path can be evaluated moving forward or backward along the graph. In [7] it has been already shown that the single-objective multidirectional modified *Physarum* algorithm is more efficient than the unidirectional algorithm when applied to small scale single-objective discrete decision making problems. This thesis will be demonstrated for the multi-objective algorithm in Sect. 4 solving a series of test cases. Symmetric Traveling Salesman and Vehicle Routing Problems (TSP and VRP), introduced in Sect. 3, were chosen as representative examples of the above type of decision making problems, here called reversible decision-making problems, i.e. problems in which a decision can be taken either moving forward or backward along the graph, as explained in Sect. 2. Simulations on large scale TSP and VRP (50 to 200 cities) are currently underway for both single-objective and multi-objective *Physarum* algorithm. First results on large scale single-objective problems are in line with previous results on small scale problems.

2 Biology and Mathematical Modeling

Physarum Polycephalum is a large, single-celled amoeboid organism that exhibits intelligent plant-like and animal-like characteristics. Its main vegetative state, the *plasmodium*, is formed of a network of veins (*pseudopodia*). The stream in these tubes is both a carrier of chemical and physical signals, and a supply network of nutrients throughout the organism [10]. *Physarum* searches for food by extending this net of veins, whose flux is incremented or decremented depending on the food position with reference to its centre. The longest is the path connecting the centre with the source of food, the smallest is the flux.

2.1 Problem Formulation: Multi-Objective Discrete Decision Making

Given a solution j to a discrete multi-objective decision making problem P, with cost vector $\mathbf{s}^j = [s_1^j, s_2^j, ..., s_n^j]$ and a solution i with cost vector $\mathbf{s}^i = [s_1^i, s_2^i, ..., s_n^i]$, the solution j dominates i if $s_k^j < s_k^i$ for all the $k = 1, 2, ..., n$. The relation $\mathbf{s}^j \prec \mathbf{s}^i$ states that $\mathbf{s}^j$ dominates $\mathbf{s}^i$. The dimension of the vector $\mathbf{s}^j$ expresses the number of evaluating criteria for a solution j. The cost vector represents the cost associated with a decision. A general problem in discrete multi-objective optimisation is to find the feasible non dominated solutions to the given discrete multi-objective decision making problem P. Following the theory developed in [14], it is possible to associate a scalar dominance index $I_d(\mathbf{s})$ to each solution. The lower is the index, the better is the solution: if one consider the set of solutions $S = \{\mathbf{s}^j, \mathbf{s}^i, \mathbf{s}^k\}$ where $\mathbf{s}^j \prec \mathbf{s}^i \prec \mathbf{s}^k$, the set of associated scalar indexes will be $I_d = \{I_d(\mathbf{s}^j) = 0, I_d(\mathbf{s}^i) = 1, I_d(\mathbf{s}^k) = 2\}$. All the non-dominated solutions in a general set S form the set:

$$PF = \{\mathbf{s} | I_d(\mathbf{s}) = 0\} \tag{1}$$

which is called Pareto front. Therefore, the solution of the problem in P translates into finding the elements of PF.

2.2 Multiple Direction Growing Decision *Physarum* Graphs

A reversible discrete decision problem can be modeled using a symmetric directed graph. The symmetry of a decision that induces a change from a state a to state b indicates here that the decision that brings back from b to a exists and can be evaluated. Not necessarily the two symmetric decisions have the same cost. The symmetric directed graph can be seen as the superposition of two directed graphs (direct-flow, *DF*, and back-flow, *BF*, graphs) whose nodes are coincident and edges have opposite orientation. In so doing, the decision between state a and b has a forward link a to b and a superposed backward link b to a. It is assumed that the first decision node is the heart of a growing *Physarum* in *DF*, and the end decision node the heart of a growing *Physarum* in *BF*. The two *Physarum* are supposed able to incrementally grow the decision graph in the two directions by extending their net of veins. A multiple direction growing decision *Physarum* graph is obtained. In the example mentioned before, the *Physarum* working in DF would build its graph by creating arcs that move from a to b. If the graph was traversed by a virtual agent, the agent would walk along an arc from a to b. The other *Physarum* would build its graph in the opposite direction then walking along each arc from b to a. The result is a graph where both nodes and links are incrementally built by two expanding *Physarum*. A mesh network topology both in *DF* and *BF* for the graph was chosen for the graph modeling of multi-objective TSP and VRP problems.

Multi-objective multidirectional incremental Modified *Physarum* algorithmic Using the Hagen-Poiseuille law, the flux through the net of *Physarum* veins is [12, 5, 10, 11]:

$$Q_{ij} = \frac{\pi r_{ij}^4}{8\mu}\frac{\Delta p_{ij}}{L_{ij}} \tag{2}$$

where Q_{ij} is the flux between i and j, μ is the dynamic viscosity, r_{ij} the radius, L_{ij} the length and Δp_{ij} the pressure gradient. In a multi-objective algorithm the value of the length can be substituted with the scalar dominance index, see Sect. 2.1, associated with a vein: the cost vectors associated with each veins that connect a node N with other nodes N_i can be compared and the dominance indexes can be evaluated. This strategy has been previously examined in [2] for Ant Colony Optimisation (ACO). Another strategy, as reported in [2], could be the use of several fluxes structures ([2] refers to pheromones), one for each objective. The choice between the two strategies was made based on a criterion of simplicity: first strategy is easier to implement and, since no multi-objective *Physarum* algorithms have been previously proposed, there were no reason in using a more complicated approach. Diameter variations allow a change in the flux. Veins' dilation due to an increasing number of nutrients flowing can be modeled using a monotonic function of the flux:

$$\left.\frac{d}{dt}r_{ij}\right|_{dilation} = f\left(Q_{ij}\right) \tag{3}$$

where $f(0) = 0$, i.e. linear, sigmoidal, etc. Veins' contraction due to evaporative effect can be assumed to be linear with radius:

$$\left.\frac{d}{dt}r_{ij}\right|_{contraction} = -\rho r_{ij} \tag{4}$$

where $\rho \in [0, 1]$ is defined evaporation coefficient. The probability associated with each vein connecting i and j is then computed using a simple adjacency probability matrix based on fluxes:

$$P_{ij} = \begin{cases} \frac{Q_{ij}}{\sum_{j\in N_i} Q_{ij}} & if\ j \in N_i \\ 0 & if\ j \notin N_i \end{cases} \tag{5}$$

where N_i is the set of neighbour for i.

An additive term in the veins' dilation process, whose first main term is expressed in Eq. (3) was added in the algorithm and takes inspiration from the behavior of the amoeba *Dictyostelium discoideum* [8]. This dilation is:

$$\left.\frac{d}{dt}r_{ij_{best}}\right|_{elasticity} = GFr_{ij_{best}} \tag{6}$$

where GF is the growth factor and $r_{ij_{best}}$ the veins' radius of the best chains of veins, i.e. the veins that form the paths in the decision graph that are in the current calculated Pareto front. The incremental growth of decision network in multiple directions is then based on a weighted roulette. Nutrients inside

Table 1: Input parameters for the modified *Physarum* solver.

m	Linear dilation coefficient, see Eq. (8).
ρ	Evaporation coefficient, see Eq. (4).
GF_{ini}	Initial growth factor, see Eq. (6)
N_{agents}	Number of virtual agents.
p_{ram}	Probability of ramification, see Sect. 2.2.
α	Weights on ramification, see Eq. (7).
r_{ini}	Initial radius of the veins.

Algorithm 1 Multidirectional incremental modified Physarum solver

```
initialize $m$, $\rho$, $GF_{ini}$, $N_{agents}$, $p_{ram}$, $\alpha$, $r_{ini}$
generate a random route from start to destination both in $DF$ and $BF$
for each generation do
    for each virtual agent in all directions ($DF$ and $BF$) do
        if current node $\neq$ end node then
            if $rand \leq p_{ram}$ then
                using Eq. (7) create a new path, building missing links and nodes
                update scalar dominance indexes, see Sect. 2.1
            else
                move on existing graph using Eq. (5).
            end if
        end if
    end for
    look for possible matchings
    update Pareto front
    contract and dilate veins using Eqs. (3), (4), (6)
    update fluxes and probabilities using Eqs. (2), (5)
end for
```

veins are interpreted as virtual agents that move in accord with adjacency probability matrix in Eq.(5). Once a node is selected, there is an *a priori* probability p_{ram} of ramification towards new nodes that are not yet connected with the actual node. If ramification is the choice, a weighted roulette, based on objective function evaluations, helps the *Physarum* with the selection and construction of a new link. The probability of a new link construction from the current node c to a new possible node $n_i \in N$, where N is the set of new possible decisions, is here assumed to be inversely proportional to the cost I_{cn_i} of the decision between c and n_i, i.e. the dominance scalar index associated with the decision:

$$p_{cn_i} \propto \frac{1}{I^{\alpha}_{cn_i}} \tag{7}$$

where α is a weight. Once a new link is built, a complete decision path is constructed (creating other links if necessary). Assuming one *DF* and one superposed *BF Physarum*, as explained in the previous paragraph, a matching condition can be then defined. If an arc connecting two nodes that belong to *DF* and *BF Physarum* respectively, exists or can be created, it is traversed by the agent and becomes part of both the *DF* and the *BF*. The matching strategy implemented in the multi-objective modified *Physarum* solver follows an elitist criterion where at each generation a joint path is selected if and only if its total cost vector is not dominated by the previous joint paths selected during the same generation. The main parameters of the modified *Physarum* solver are listed in Table 1. The initial radius of the veins r_{ini} is

Table 2: Values used as input parameters - TSP test case (first row) and VRP test case (second row).

m	ρ	GF_{ini}	N_{agents}	p_{ram}	α	r_{ini}
5×10^{-5}	$\frac{1 \times 10^{-5}}{N_{agents}}$	5×10^{-3}	100	0.8	0	1
5×10^{-5}	$\frac{1 \times 10^{-5}}{N_{agents}}$	1×10^{-3}	150	0.8	0	1

always set equal to 1 in the simulations presented in this paper. The pseudocode of the multidirectional incremental modified *Physarum* solver is provided in Algorithm 1. A unidirectional algorithm is a special case of multidirectional algorithm, obtained by freezing the BF: flux and graph growth are allowed in only one direction. Simulations on selected test cases (see Sect. 4) were carried out adding in the *Physarum* software a routine for the adaptive control of the growth factor GF. This control was introduced in order to incrementally boost the effect of GF during a simulation, driving exploring agents towards best veins. Simulations showed that the adaptive control of GF helps the convergence of the algorithm towards optimal solution. Given an initial value for the growth factor GF_{ini}, GF is incremented by a fixed percentage σ after every generation. If the highest probability p_{best}^{PF} associated with the paths in the calculated Pareto front so far is higher than a fixed value p_{lim}^{low} , the increment is set to zero. Then, if p_{best}^{PF} exceeds a value p_{lim}^{high} , GF is set equal to GF_{ini} and veins are dilated and contracted to their initial value. In the present paper is assumed $\sigma = 0.01$, $p_{lim}^{low} = 10^{-4}$, $p_{lim}^{high} = 0.85$ for the Vehicle Routing Problem test case and $p_{lim}^{high} = 0.95$ for the Traveling Salesman Problem test case.

Considerations on the algorithm The set of Eqs. (2)-(5) can be implemented as in the following. In accordance to Eq. (2), flux in each vein is proportional to the radius and inversely proportional to the length (the scalar dominance index in a multi-objective problem). These two main parameters are taken into account in the algorithm. Once a vein is selected by a virtual agent in a generation, its radius is incremented using Eq. (3). In the present work, a linear function with respect to the product between the radius $r_{ij}^{(n)}$ of the veins traversed by agent n, and the inverse of the sum of dominance indexes ($I_{tot}^{(n)}$, see Sect. 2.1), associated with each arc of the decision taken by agent n, will be used for the veins' dilation:

$$\left. \frac{d}{dt} r_{ij}^{(n)} \right|_{dilation} = m \frac{r_{ij}^{(n)}}{I_{tot}^{(n)}} \tag{8}$$

where the coefficient m is the linear dilation coefficient. Evaporation is taken into account using Eq. (4) for each agent. Fluxes are then calculated using Eq. (2) and probabilities are updated in accordance with Eq. (5).

3 Application to Traveling Salesman and Vehicle Routing Problems and Benchmark

In single objective optimisation the Traveling Salesman problem, TSP, is the problem of finding the shortest tour that visit each city of a given set S of n cities. In the multi-objective optimisation case considered in this paper the cost function to be minimized is a vector of two values: the total length $L_{tot} = \sum_j L_j$ and the total road traffic $T_{tot} = \sum_j T_j$ of each tours, where $j = 1, ..., n$ is the index that identifies each part of the tour. The road traffic is here assumed to be inversely proportional to the length $T_j = 1/L_j$. The shorter is the tour, the higher is the probability that the tour is chosen by drivers, increasing the road traffic. The total road traffic $T_{tot} = \sum_j T_j$ will be called Road Traffic Index in the following. Although conflicting criteria, both length and road traffic in a tour have to be minimized.

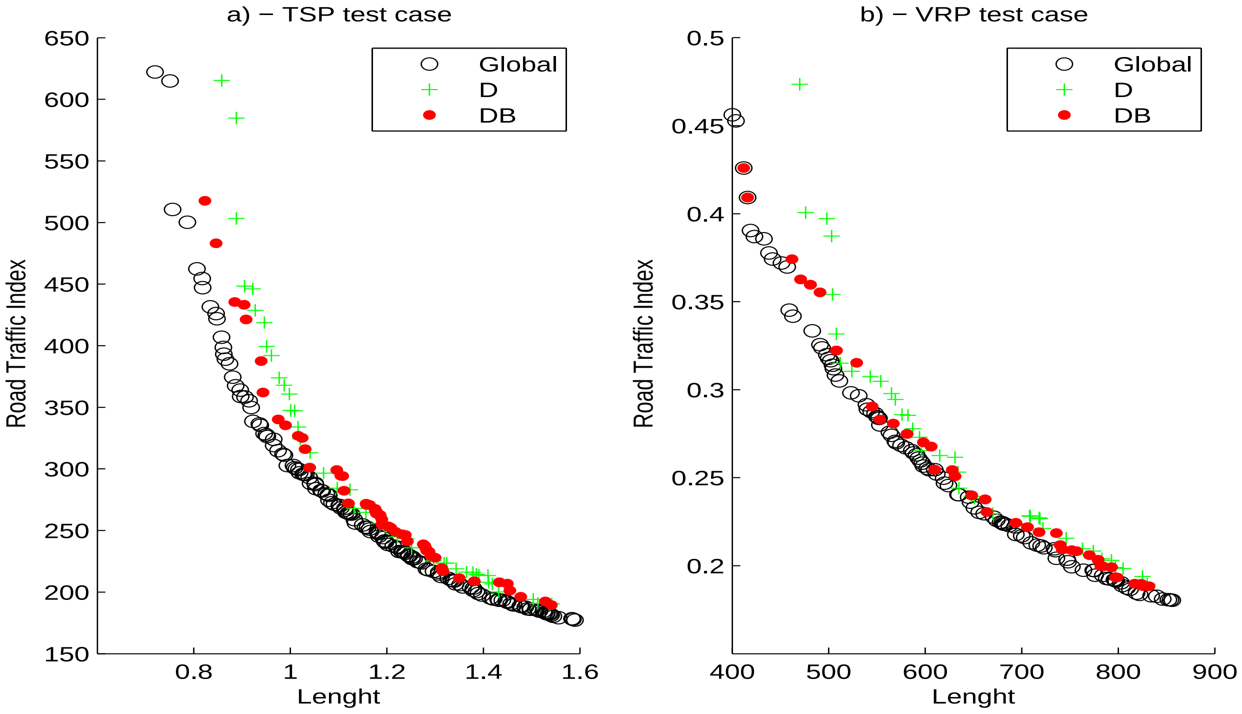

Figure 1: Pareto front - TSP test case at 650000 function evaluations, (a) - VRP test case at 300000 function evaluations, (b).

TSPLIB [13] was used to benchmark the proposed *Physarum* algorithm, developed in Matlab® R2010b, on the TSP problem. In Sect. 4 are reported the results obtained by applying the multi-objective multi-directional *Physarum* solver to test case *Ulysses16* that was modified adding the road traffic to the cost function and normalising the length with a factor equals to 10000.
The multi-objective Vehicle Routing Problem, VRP, considered in this paper is a similar problem. Given a set of n cities with a demand k, whose reciprocal distance L_j and road traffic $T_j = 1/L_j$ are known, v vehicles of capacity c, d depots located in fixed cities, the VRP is the problem of delivering goods located in the depots using a defined amount of vehicles with finite capacity. The goal is to satisfy the demand of each city minimizing the cost functions, i.e. the distance and road traffic. VRP reduces to a TSP if there is only one vehicle with infinite capacity. When the modified *Physarum* algorithm is applied to VRP, a probability skew factor ψ is included in the algorithm. If an agent is not obliged to go to depot, the probability to reach the depot is lowered of a factor $(1 - \psi)$. Other probabilities are then risen of a same value in order to have the sum of probabilities equals to 1. The skew factor ψ is introduced in the model to avoid frequent returns to depot in the decision sequences and is here set equals to 0.5. The *Physarum* solver applied to VRP was tested on a map of 9 cities plus one depot. The map is built using 9 Italian cities (Firenze, Livorno, Montecatini, Pistoia, Prato, Montevarchi, Arezzo, Siena, San Gimignano), with a city considered the depot (Ponsacco). The Euclidean distance in kilometers was used. VRP parameters were set to $n = 9, k = cost = 1, v = 1, c = 4, d = 1$, i.e. one vehicle with capacity equals to 4, one depot and a constant demand equals to 1.

3.1 Testing Procedure

The testing procedure proposed in [14] was used in this paper. Both the metrics M_{spr} and M_{conv}, defined in [14], should be low for a good estimate of the global calculated Pareto front. The indexes of performance

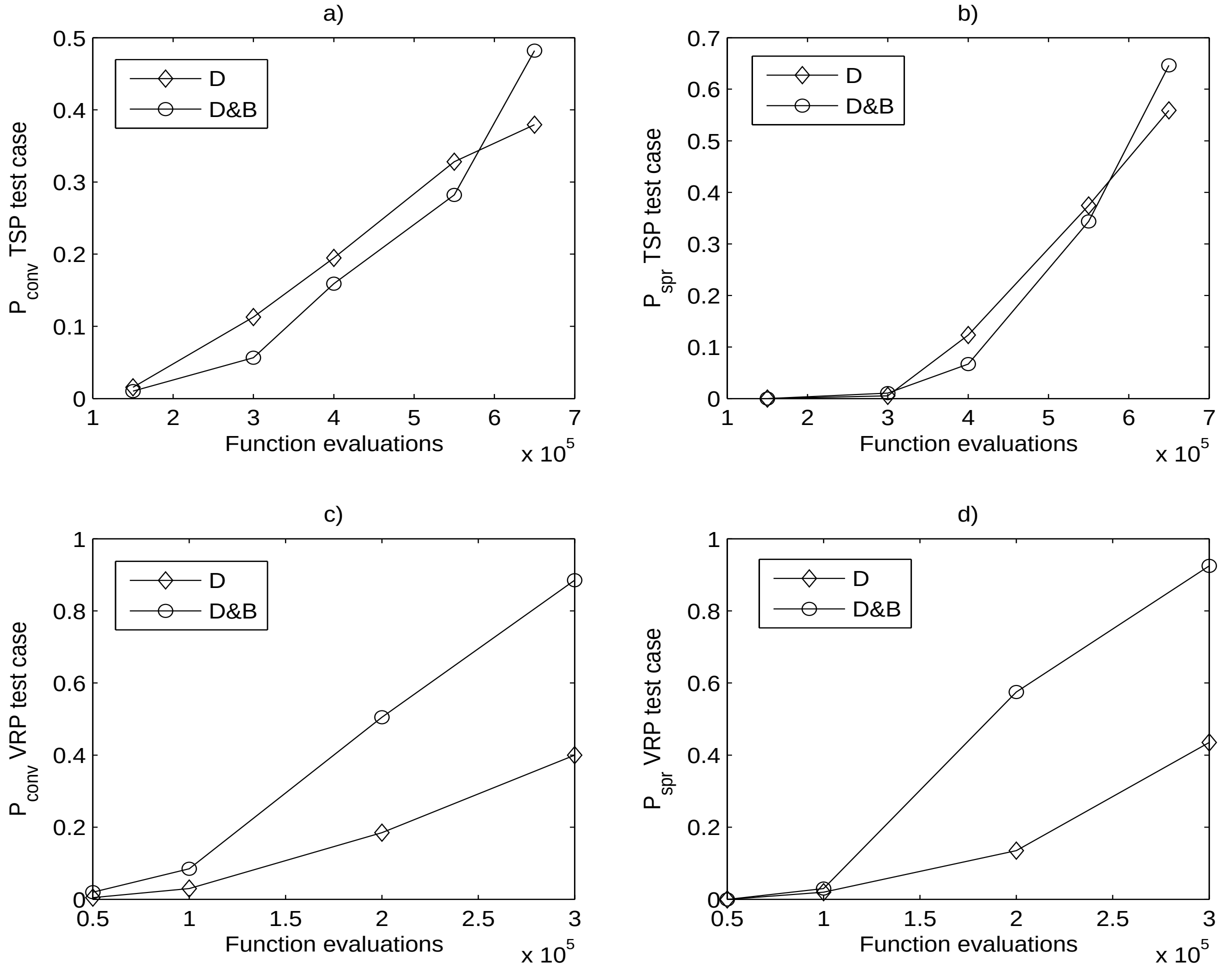

Figure 2: Variation of the indexes of performance p_{conv} and p_{spr} with the number of function evaluations - TSP test case, (a) and (b) - VRP test case, (c) and (d).

$p_{conv} = P(M_{conv} < tol_{conv})$ and $p_{spr} = P(M_{spr} < tol_{spr})$ will be used to explore the efficiency of the algorithm and to compare the multidirectional and the unidirectional versions. Given n repeated runs, p_{conv} is the probability that M_{conv} achieves a value less than tol_{conv}, while p_{spr} is the probability that M_{spr} achieves a value less than tol_{spr}. 200 runs are sufficient in order to obtain an error $\leq 5\%$ with a 95% of confidence [14]. For the TSP test case the tolerances tol_{conv} and tol_{spr} are set equal to 0.0465 and 0.045 respectively, while for the VRP test case the tolerances tol_{conv} and tol_{spr} are set equal to 0.030 and 0.035 respectively.

4 Results

The multi-objective multidirectional modified *Physarum* solver, named D&B in the following, was compared against a multi-objective unidirectional modified *Physarum* solver, named D.

The two algorithms were applied to the modified symmetric traveling salesman problem test case *Ulysses16* and to the vehicle routing problem test case, described in Sect. 3. The values used as input parameters in the simulations, chosen after a series of trials, are listed in Table 2. Selected ones showed best performance. Simulations were carried out on a 64-bit OS Windows 7 Intel® Core™2 Duo CPU E8500 3.16GHz 3.17GHz.

In both the test cases the true global Pareto front was unknown. In order to obtain a global Pareto front all the runs (1600 for the VRP test case and 2800 for the TSP test case) of the multidirectional and unidirectional algorithms were used: two global Pareto fronts for both VRP and TSP test cases were built using all the solutions found by the algorithms. In Fig. 1 the global Pareto fronts for TSP and VRP are showed. The figure reports also an example of Pareto front found by unidirectional (D) and multidirectional (DB) algorithms, for both TSP and VRP test cases.

Fig. 2 shows the variation of the indexes of performance p_{conv} and p_{spr} with the number of function evaluations for the TSP test case ((a) and (b)) and for the VRP test case ((c) and (d)). A function evaluation is defined as the call to the objective function, i.e. each arc selected by the virtual exploring agents (see Sect. 2.2) is considered a function evaluation.

Results for the VRP test case, as shown in Fig. 2 (c) and (d), demonstrate that the multi-objective multidirectional modified *Physarum* algorithm with matching ability (D&B) provides higher indexes of performance p_{conv} and p_{spr}, than the multi-objective unidirectional modified *Physarum* algorithm (D), at all the function evaluations limit. This gain is up to approximately 50% for the p_{spr} and p_{conv} at 300000.

The results for TSP test case, reported in Fig. 2 (a) and (b), show that the multidirectional algorithm provides better performance after 650000 function evaluations and the gain is up to 10% for both p_{spr} and p_{conv} at 600000. The behaviour of the indexes of performance for this multi-objective instance is similar to the behaviour of the index of performance in [7] for the same TSP test case with single-objective: the unidirectional algorithm tends to have a better performance during the early stage of the simulation, then the performance of the multidirectional algorithm exceeds the performance of the unidirectional.

These results are quite interesting and prove the initial assumption that building decision sequences in two directions and adding a matching ability is an advantageous choice if compared with the choice of building decision sequences in only one direction in the solution of multi-objective discrete decision making problems. The two *Physarum* can evaluate each step of the decision sequence from two directions and create joint paths: this forward and backward decision making process improves the performance of the algorithm.

4.1 Conclusion

This paper proposed a multi-objective multidirectional incremental modified *Physarum* solver for multi-objective discrete decision making problems. The algorithm showed the ability to solve multi-objective problems in combinatorial optimization, i.e. symmetric traveling salesman and vehicle routing problems, that were selected as representative examples of multi-objective reversible decision making problems. Simulations on selected test cases proved that a multidirectional approach with matching ability performs better than a unidirectional one when applied to small scale multi-objective reversible discrete decision making problems. This result is in line with the results showed in [7] for single-objective discrete decision making using a multidirectional modified Physarum algorithm. The multidirectional decision making process enhances the performance of the multi-objective solver: this gain is up to 50% (based on the indexes of performance proposed in Sect. 3.1) for the VRP test case. Simulations on large scale TSP and VRP (50 to 200 cities) are currently underway for both single-objective and multi-objective *Physarum* algorithm. First results on large scale single-objective problems are in line with previous results on small scale problems.

References

[1] A. Adamatzky, G.J. Martinez, S.V. Chapa-Vergara, R. Asomoza-Palacio, C.R. Stephens, "Approximating Mexican Highways with Slime Mould", *Natural Computation*, in press, 2011.

[2] I. Alaya, C. Solnon, K. Ghedira, "Ant Colony Optimization for Multi-objective Optimization Problems",*19th IEEE International Conference on Tools with Artificial Intelligence*, Patras, Greece, v. 1, p. 450-457, 2007.

[3] M. Dorigo, L.M. Gambardella, “Ant Colonies for the Traveling Salesman Problem”, *BioSystems*, v. 43, p. 73-81, 1997.

[4] M. Dorigo, V. Maniezzo, A. Colorni, “The Ant System: Optimization by a colony of cooperating agents”, *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, v. 26(1), p. 29-41, 1996.

[5] D.S. Hickey, L.A. Noriega, “Insights into Information Processing by the Single Cell Slime Mold Physarum Polycephalum”, *UKACC Control Conference*, Manchester, UK , September 2-4, 2008

[6] M. Lopez-Ibanez, “Multi-objective Ant Colony Optimization”,*Diploma thesis*, Intellectics Group, Computer Science Department, Technische Universitat Darmstadt, Germany, 2004.

[7] L. Masi, M. Vasile, “A Multidirectional Modified Physarum Solver for Optimal Discrete Decision Making”, *Proceedings of International Conference on Bio-Inspired Optimization Methods and their Applications, BIOMA*, Bohinj, Slovenia, May 24-25, 2012.

[8] D.R. Monismith Jr., B.E. Mayfield, “Slime Mold as a Model for Numerical Optimization”, *2008 IEEE Swarm Intelligence Symposium*, St. Louis MO, USA, September 21-23, 2008.

[9] T. Nakagaki, H. Yamada, A. Toth, “Maze-Solving by an Amoeboid Organism”, *Nature*, v. 407, p. 470, 2000.

[10] A. Tero, K. Yumiki, R. Kobayashi, T. Saigusa, T. Nakagaki, “Flow-Network Adaptation in Physarum Amoebae”, *Theory in Biosciences*, v. 127(2), p. 89-94, 2008.

[11] A. Tero, R. Kobayashi, T. Nakagaki, “Physarum Solver: a Biologically Inspired Method of Road-Network Navigation”, *Physica: A Statistical Mechanics and its Applications*, v. 363(1), p. 115-119, 2006.

[12] A. Tero, S. Takagi, T. Saigusa, K. Ito, D.P. Bebber, M.D. Fricker, K. Yumiki, R. Kobayashi, T. Nakagaki, “ Rules for Biologically Inspired Adaptive Network Design”, *Science*, v. 439, p. 327, 2010.

[13] TSPLIB, library of instances for Traveling Salesman and Vehicle Routing Problems, Ruprecht Karls Universitaet Heidelberg, URL: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.

[14] M. Vasile, F. Zuiani, “A Hybrid Multiobjective Optimization Algorithm Applied to Space Trajectory Optimization”, *World Congress on Computational Intelligence*, Barcelona, Spain, July 18-23, 2010 .