Pearl Hunter: A Cross–domain Hyper–heuristic that Compiles Iterated Local Search Algorithms

7th September 2011, Nottingham, UK

Fan Xue, CY Chan, WH Ip, CF Cheung
Department of Industrial & Systems Engineering
Hong Kong Polytechnic University

The Hong Kong Polytechnic University
香港理工大學

Department of Industrial and Systems Engineering
工業及系統工程學系

Outline

1 Introduction

2 Pearl Hunter

3 Training and Validation on HyFlex

4 Migrating to Quadratic Assignment

5 Discussion and conclusion

Ancient stories (intelligence & optimization)

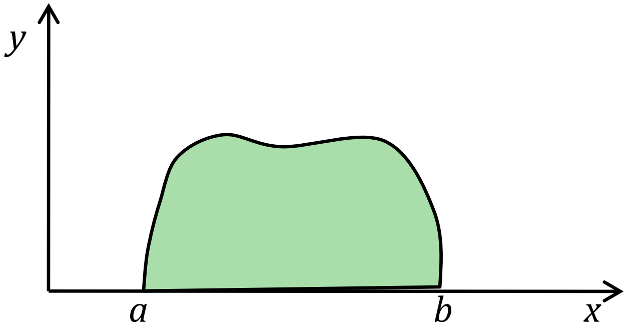
- Carthaginian bull's hide
  - (In 810s B.C.,) … Elissa (Dido) had an bull's hide cut into strips and lay them out end-to-end in a *crescent* circumscribing a sizeable area of land. This ox-hide enclosed area was known as Carthage.

Ruins of Carthage (from Wikipedia)

- Dido's problem
  - max $\int_a^b f(x)dx$

    s.t. $\int_a^b \sqrt{1+(f'(x))^2}dx = \mathrm{Len}_{\mathrm{oxhide}}$

    $f(a) = f(b) = 0$
  - The correct answer is Dido's *semicircle*, and d(a, b) is the diameter.

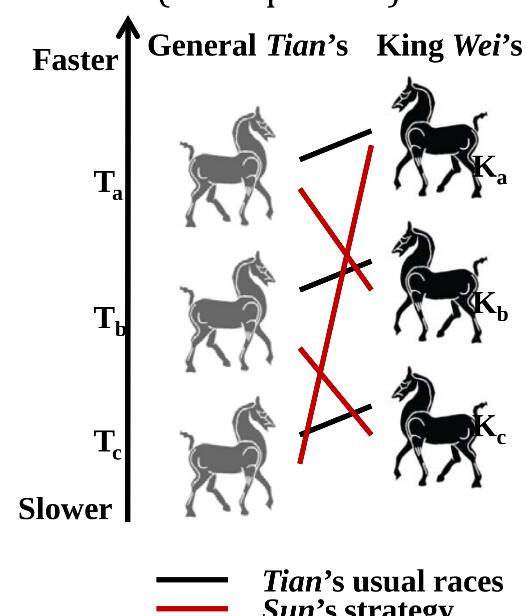


Dido's problem

Ancient stories (intelligence & optimization)

- *Tian Ji*'s racing horses
  - "(In 340s B.C.,) General *Ji Tian* of Kingdom *Qi* raced horses with other members of royal family several times. His guest *Bin Sun* (author of *Sun Bin's Art of War)* found *Tian*'s 3 horses covered 3 levels and were not much inferior in races…" (Sima, 91bc)
- *Sun*'s strategy
  - Displacement: sent *Tian*'s inferior horse (T_c) to race in the name of the best (T_a), T_a to race in the name of the average(T_b), T_b to race in the name of the inferior (T_c).
  - *Tian* and King *Wei* of *Qi* had a horseracing. *Tian* won 2/3 races, and won a prize of about 500 oz copper. *Sun* became the military counselor of *Qi*.

Ruins of Ancestral Temple of Qi (from qidu.net)



Hyper-heuristics

- Enormous of optimization methods have been proposed so far. Four issues have been concerned:
  - *Effectiveness* able to find highly satisfactory solutions,
  - *Efficiency* with quick running,
  - *Easiness* (IMO) easy to understand and deploy, and
  - *Portability* scalable to *different domains* and datasets.
- Machine Learning: main source of the portability power
- Hyper-heuristics $\subseteq^{(?)}$ Metaheuristics
  - Hyper-heuristics *select* or *generate* heuristics via *online* or *offline* learning, to combine the strength and to compensate the weakness of each "low-level" heuristic (like *Sun*'s?), if each heuristic has its own strength and weakness.

Low-level heuristics

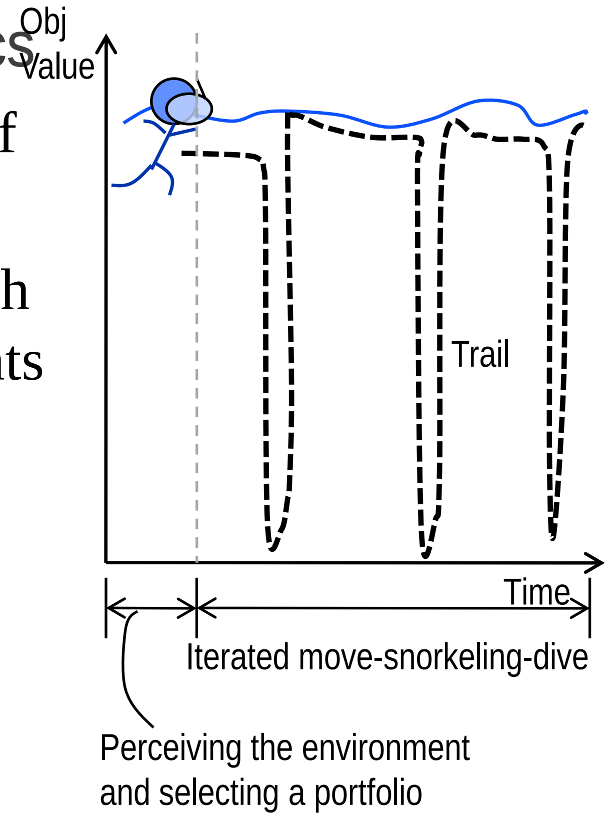
Pearl Hunter: an inspiration

- *Pearl hunting* is an out-of-date diving activity of retrieving pearls from oysters. Can still be found in:
  - Some Aisa tourist sites,
  - Virtual games.
- In a search perspective, pearl hunting consists of repeated
  - *diversification* (surface and change target area)
  - *intensification* (dive and find pearl oysters).
- Pearl hunting is in the paradigm of Iterated Local Search (Lourenço *et al*, 2003).

Pearl hunting in World of Warcraft

Pearl Hunter: a hyper-heuristic imitation

- Basic actions made of low-level heuristics
  - *Snorkeling*: local search with a low "depth of search", stops after any improvements
  - *Deep dive* (SCUBA): local search with a high "depth of search", till no further improvements
    - Why two intensifications? $N_{\text{snorkeling}}/N_{\text{dive}}$?
  - *Surface moves*: non-local-search heuristics
    - *Crossover* (MI): 2 or more input solutions
    - *Mutation* (SI): 1 input solution

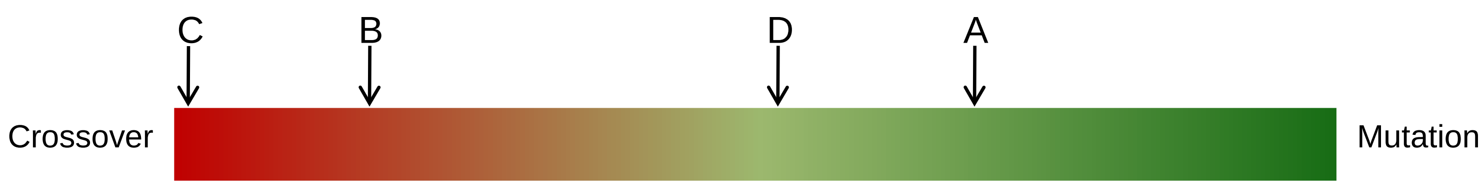


- "Environment":
  - *Shallow water*, where local search is useless
  - *Sea trench*, where local search costs too much time
  - *Default, otherwise*

Pearl Hunter: a hyper-heuristic imitation (Continued)

- Pearl Hunter can drop a *Buoy* at the depth of first deep dive, to escape from local optimum by mutations (SIs).
- Four running modes (portfolios) of moves:
  - **A**: all moves averagely, with a *Buoy* mark
  - **B**: *crossover* with a *Buoy* mark (triggering a few mutations)
  - **C**: *crossover* only, no mutation, no *Buoy*
  - **D**: Sea trench mode, all surface moves averagely, no *Buoy*.

  Moves are subject to online pruning.



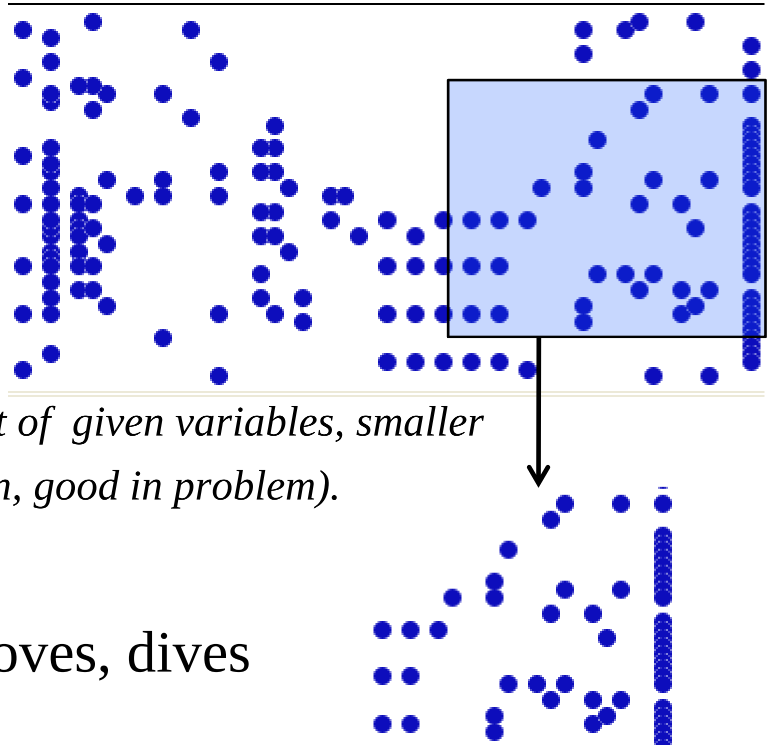
Anchors of the modes

- Other tricks:
  - tabu lists (memory), “mission restarts” (go to new areas)

Pearl Hunter: a hyper-heuristic imitation (Continued)

- 1)Selecting low-level heuristics and 2)determining one mode after the "perceiving" period (classification)
  - Off-line learning
    - Rule induction ♥
    - …
  - Online learning
    - Full problem
    - A division of problem ♥

A "cropped" problem with a subset of given variables, smaller but "<u>keeps flavor</u>" (good in division, good in problem).



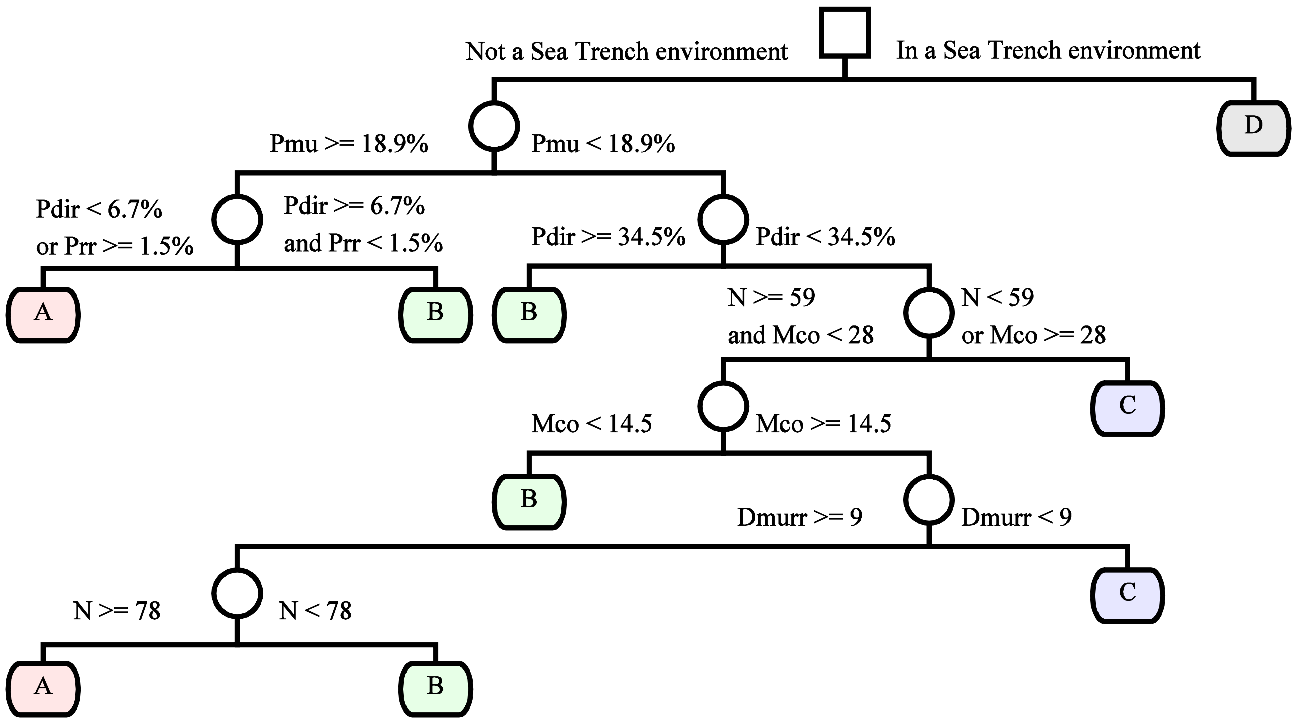
- Attributes for offline learning:
  - Suboptimal solutions found by moves, dives
  - Restarts, MI vs SI, …

Example: a division of a TSP (u159)

HyFlex and CHeSC

- HyFlex (Hyper-heuristics Flexible framework) is a java cross-domain platform (Burke *et al*, 2011)
  - 6 domains, 4 public (training domain) and 2 hidden
  - “Black-box” low-level heuristics in 4 categories:
    - Crossover, Mutation, Ruin-recreate, and Local search
  - Parameters to control low-level heuristics :
    - “Intensity" of mutations, and “depth of local search”
- CHeSC 2011 is the first Cross-domain Heuristic Search Challenge on HyFlex. (http://www.asap.cs.nott.ac.uk/chesc2011/)
- Pearl Hunter was ranked in CHeSC:
  - 4th out of 20 entries overall,
  - 1st out of 20 entries in the hidden domains.

HyFlex and CHeSC: BF-Tree obtained by offline learning (by Weka v3.5)



- D_{murr}: Depth of the mission in the Mutation and Ruin-recreate test,
- M_{co}: Number of missions completed in the Crossover test,
- N: Number of sub-optimal solutions found in total,
- P_{dir}: Percent of sub-optimal solutions found right after some moves (before any dive),
- P_{mu}: Percent of sub-optimal solutions found in iterations started with Mutation moves,
- P_{rr}: Percent of sub-optimal solutions found in iterations started with Ruin-recreate moves,

Tests on personnel scheduling: beyond the 600s time limit of CHeSC

- On large-scale personnel scheduling problems,
  - Running time was increased to 10 hours (normalized to P4 3GHz),
  - Same decision tree
- New best known solutions:

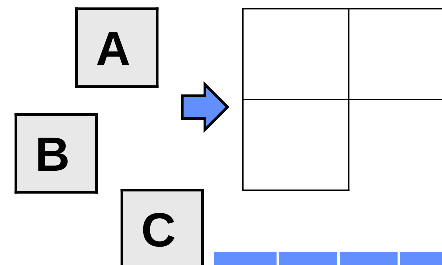
Instance	Men | days	Time (h)	Result	Prev BK*	% improved
CHILD-2A	41 | 42	10	*1,095*	1,111	1.4
ERRVH-A	51 | 42	10	*2,142*	2,197	2.5
ERRVH-B	51 | 42	10	*3,121*	6,859	54.5
MER-A	54 | 42	10	*9,017*	9,915	9.1

* Best known values were collected from http://www.cs.nott.ac.uk/~tec/NRP/misc/NRP_Results.xls

- A possible reason
  - A new “vertical” swap concept first implemented in low-level heuristics on HyFlex

QAP: Another test domain

- Quadratic assignment problem (QAP)
  - $\sum_{a,b \in F} \omega_{a,b} \cdot d_{A(a),A(b)}$ where A is assignment.
  - Example: place *N* facilities in a grid of cellular manufacturing (facility layout problem).
  - NP-hard
- Coded as a new domain on HyFlex
- Low-level heuristics implemented
  - Crossover heuristics
    - Partially Matched Crossover (Chan and Tansri,1994)
    - Order Crossover (Chan and Tansri,1994)
    - A voting recombination crossover



w	A	B	C
A	0	3	1
B	3	0	6
C	1	6	0

QAP: Low-level heuristics (continued)

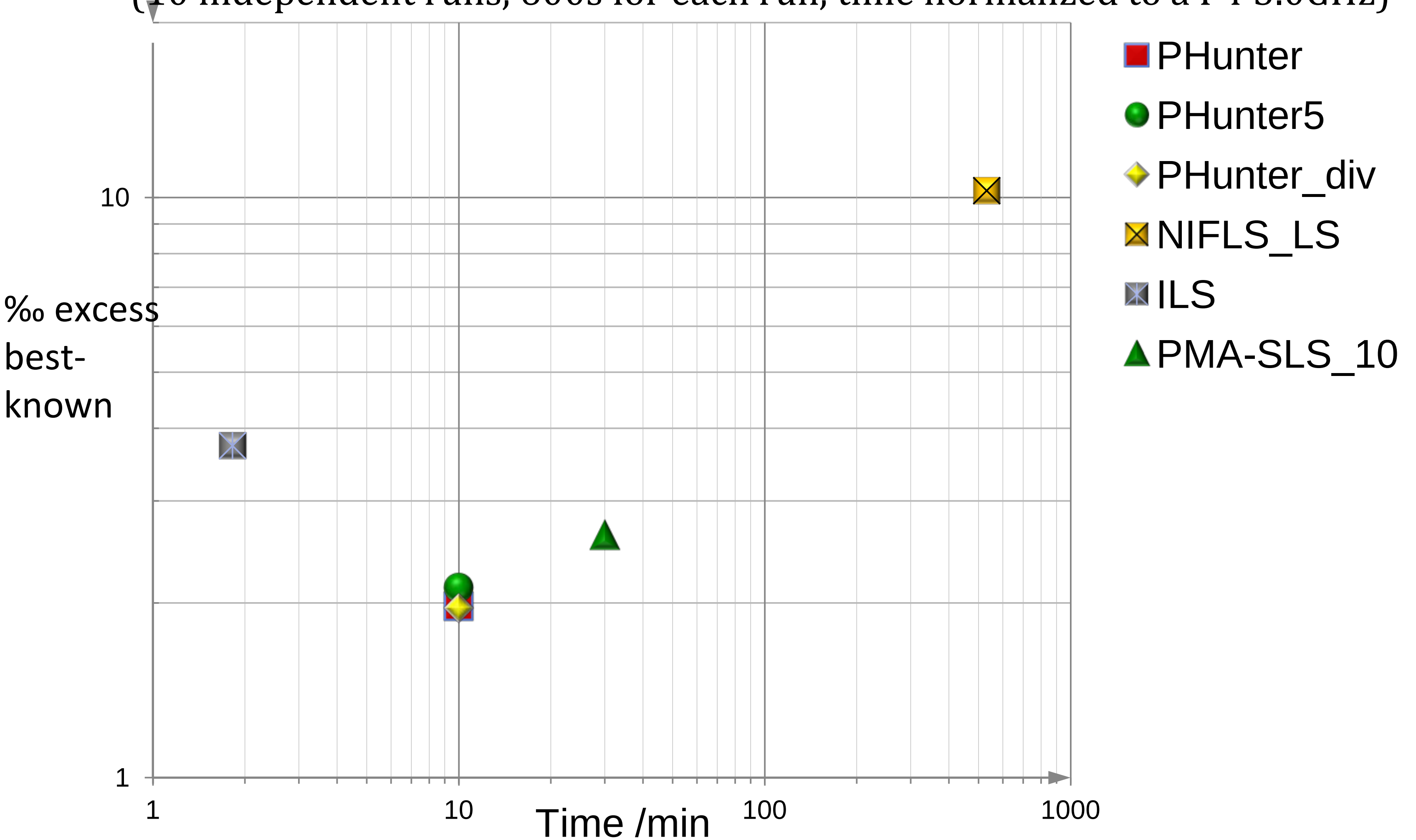
- Mutation heuristics
  - Random swaps
  - Shifting mutation (PSSC Lab, 2005)
  - Spiral reassignment (Yaman *et al*, 1993)
- Ruin-recreate heuristics
  - Chan's heuristic (Chan *et al*, 2002)
  - GRASP (greedy randomized adaptive search procedure, Feo and Resende, 1995)
- Local search heuristics
  - Variable Depth Search with partial gains (Burke *et al*, 2007)
  - Tabu Search (Taillard, 1991)
- Division selection heuristic
  - Selecting flow and distance matrices with closest means and deviations from 1000 random division samples.

QAP: Experiments

- Instances
  - 10 largest from QAPLIB, Euclidean and non-Euclidean
- Algorithms
  - **PHunter**: Codes and rules for CHeSC (no modification)
  - **PHunter$_5$**: Same codes, Appended QAP to training domains (4->5)
  - **PHunter$_{div}$**: Same codes, CHeSC rules, appended a simple online mode learning via a division ($N'= 0.47*N$) : try different modes independently (5% time, pow(5%, 0.25)=0.47), chose the best one.
  - **NIFLS_LS**: Iterated local search (Ramkumar *et al*, 2009).
  - **ILS**: Iterated Local Search (Stützle, 2006)
  - **PMA-SLS$_{10}$**: Parallel Memetic Algorithm with Selective Local Search (10 islands, Tang *et al*, 2006)

QAP: Experiments (continued)

Off-the-peg Hunters versus custom-made methods on *N*=100 instances (10 independent runs, 600s for each run, time normalized to a P4 3.0GHz)



QAP: A close look of results

Average objective values excess best-known values (%)

Prob	BK	PH	PH_5	PH_{div}	$NIFLS_{LS}$	ILS	$PMA\text{-}SLS_{10}$
Sko100a	*152002*	0.0566	0.0577	0.0597	0.32	0.312	0.0663
Sko100b	*153890*	0.0248	0.0130	0.0164	0.49	0.5068	0.0636
Sko100c	*147862*	0.0122	0.0104	0.0114	0.34	0.6023	0.0226
Sko100d	*149576*	0.0602	0.0602	0.0850	0.59	0.021	0.0706
Tai100a	*21052466*	0.6999	0.6654	0.6464	1.83	0.6933	1.5684
Tai100b	*1.19E+09*	0.4908	0.6390	0.4966	3.36	---	0.0048
Tai150b	*4.99E+08*	0.6080	0.6080	1.440	---	0.095	---
Tai256c	*44759294*	0.2984	0.2984	0.2804	0.34	---	---
Tho150	*8133398*	0.1171	0.1171	0.1684	---	0.068	0.1418
Wil100	*273038*	0.0383	0.0429	0.0565	0.26	0.1041	0.0332

Discussion

- The off-the-peg hyper-heuristics can be comparable to the custom designed metaheuristics
- Results of PH, PH_5, PH_{div} are very close
  - PH: Pearl Hunters seem portable without tweak on codes
  - PH_{div}: Capability of learning online from a proper division
- Difficulties in getting size of division *N'*:
  - *N'* could be determined by: $t_{LS}(N') / t_{LS}(N) = t_{perceiving} / t_{hunting}$
    - Proportions of local search may change, e.g., $t_{LS1}=O(N^5)$, $t_{LS2}=O(N^2)$
  - Complexity of division is still not well redressed by the equation. (NP-hardness versus polynomial algorithms)
    - *N'* too small, easy to find optimum, unable to rank heuristics;
    - $t_{perceiving} / t_{hunting}$ larger, less time for hunting.

Discussion (continued)

- Eventually a "point and shoot" hyper-heuristic software for daily use?
  - Step 1: Define variables, an objective function, and constraints,
  - Step 2: Solutions out.

Academic	Algorithm idea	Implementation details	Problem def	Engineering
Academic	Algorithm idea	Implementation details	Problem def	Engineering

  - Cross-domain crossover, mutation, local search?
- More accessible function on HyFlex?
  - Such as "Similarity between two solutions"
- "Learn-and-generate" hyper-heuristic on HyFlex?
  - Encapsulate training data <attribute1_*i*, value1_*i*, attribute2_*i*, value2_*i*, …, label> for each low-level heuristic *i*?

Conclusion

- We present a hyper-heuristic
  - Imitates pearl hunting
  - Perceives "environment" of search
  - Determines a perturbation mode by online / offline learning
  - Generates different modes of ILS
- We find the results of tests encouraging
- Possible future works
  - Other reasonable ways to classify mode online
  - Hunters can generate new low-level heuristics
    - (Custom designed for TSP) Generated an association-rules-based weighting heuristic to determine candidate set, and facilitated branch-and-bound and local search (LKH) (Xue *et al*, 2010).

References

- Burke E.K., Curtois T., Qu R., Vanden Berghe G. (2007) A Time Pre-defined Variable Depth Search for Nurse Rostering. Technical Report No. NOTTCS-TR-2007-6. University of Nottingham
- Burke, E. K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., and Qu, R. (2010) Hyperheuristics: A Survey of the State of the Art, University of Nottingham. Technical Report No. NOTTCS-TR-SUB-0906241418-2747.
- Burke, E., Curtois, T., Hyde, M., Ochoa, G., Vazquez-Rodriguez J. A. (2011) HyFlex: A Benchmark Framework for Cross-domain Heuristic Search, ArXiv e-prints, arXiv:1107.5462v1
- Chan W. M., Chan C. Y., Ip W. H. (2003). A heuristic algorithm for machine assignment in cellular layout. Comput. Ind. Eng. 44(1):49-73
- Feo T.A., Resende M.G.C. (1995) Greedy randomized adaptive search procedures. J. of Global Optimization, 6:109-133.
- Lourenço, H., Martin, O., Stützle, T. (2003) Iterated Local Search. In Glover, F., Kochenberger, G. (eds.) Handbook of Metaheuristics. Springer New York. 320-353.
- Production Scheduling and Soft-Computing Lab (2005) The Genetic Algorithm for Solving the Quadratic Assignment Problem and Continuous Problem. available at: http://peterchenweb.appspot.com/publications/sourceCodes/OpenGA/OpenGA.pdf
- QAPLIB: http://www.seas.upenn.edu/qaplib/
- Ramkumar, A. S., Ponnambalam, S. G., Jawahar N. (2009) A new iterated fast local search heuristic for solving QAP formulation in facility layout design, Robotics and Computer-Integrated Manufacturing, 25(3):620-629.
- Sima, Q. (91B.C.) Biographies of Sun Tzu and Wu Qi, in Records of the Grand Historian, Vol. 65. (In Chinese)
- Stützle, T. (2006) Iterated local search for the quadratic assignment problem, European Journal of Operational Research, 174(3):1519-1539
- Taillard E. (1991) Robust taboo search for the quadratic assignment problem, Parallel Computing 17:443-455.
- Tang, J., Lim, M. H., Ong Y. S., Er M. J. (2006) Parallel memetic algorithm with selective local search for large scale quadratic assignment problems, International Journal of Innovative Computing, Information and Control, 2(6):1399-1416
- Xue F., Chan, C.Y., Ip, W.H., Cheung, C.F. (2010) Towards a learning-based heuristic searching reform scheme, XXIV European Conference on Operational Research (EURO), Lisbon, Portugal.
- Yaman, R., Gethin, D. T., Clarke, M. J. (1993). An effective sorting method for facility layout construction. International Journal of Production Research, 31(2), 413-427.

Thank you for your attention!

E-mail addr.: mffxue@inet.polyu.edu.hk
dewolf_matri_x@msn.com

THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

Department of
Industrial and Systems Engineering
工業及系統工程學系