WHISTLING: Wasp Behavior Inspired Stochastic Sampling

Vincent Cicirello and Stephen F. Smith

The Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213
{cicirello,sfs}@cs.cmu.edu

Introduction

- **Question:** *How can we cover higher-valued points of solution-space in combinatorial domains efficiently?*
- Search heuristics can provide a basis
- Heuristics are not infallible
- We must balance adherence to heuristic against possibility of missing better solutions
- Randomization as approach to hedging on this trade-off

Dispatch Scheduling Policies

Local rules for prioritizing work on different resources and coordinating material flows

- Examples: FIFO, WSPT, ATC

Advantages:

- Simple, robust control regime

Disadvantages:

- Decisions tend to be myopic
- No one heuristic tends to dominate across varying production characteristics

Some Characteristics of Dispatch Heuristics

- Typically quite sensitive to parameter settings
  - Often tuned to individual problem instances during experimental evaluation
- Typically designed and validated under idealized modeling assumptions
  - Adapted to account for additional constraints

Research question: ***Can the performance of such decision rules be improved by adding randomness?***

Amplifying Dispatch Heuristics

Starting assumption: *We have a good heuristic, but its discriminatory power varies from context to context*

Approach: *Calibrate the degree of randomness in the heuristic's choice to the level of uncertainty in a given decision context*

Some Related Ideas:

- Limited Discrepancy Search [Harvey & Ginsberg 95]
- Depth-Bounded Discrepancy Search [Walsh 97]
- Heuristic Equivalency [Gomes, Selman, & Kautz 98]
- Heuristic-Biased Stochastic Sampling [Bresina 96]
- Random-PCP [Oddi & Smith 97], Iterative Flattening [Cesta, Oddi, & Smith 99]

Carnegie Mellon

Limited Discrepancy Search (LDS) [Harvey & Ginsberg, 1995]

- A systematic backtrack search procedure
- Iteration 0: follow search heuristic at each decision point
- Iteration *j*: systematically consider each solution trajectory with *j* **discrepancies** from the heuristic path
- Continue until feasible solution found or search-space exhausted

Carnegie Mellon

Depth-bounded Discrepancy Search [Walsh, 1997]

- Assumes heuristic's advice most fallible near root of search-space
- An iterative-deepening variation of LDS
- Iteration *j*: Perform LDS restricting discrepancies to depth *j* of search-space
- Continue until feasible solution found or search-space exhausted

Carnegie Mellon

Iterative Sampling [Langley, 1992]

- At each decision point, choose a branch of the search space at random until a leaf-node is reached.
- If an infeasible solution is found, return to root of search space and iterate.
- If a feasible solution is found and if this solution is better than the best found so far, then replace the best found solution with this solution. Return to root and iterate.
- A rather naïve approach:
  - **Assumes a large number of feasible solutions**
  - **Assumes a large number of "good" solutions**

Carnegie Mellon

Heuristic-Biased Stochastic Sampling [Bresina,1996]

- At each decision point, rank order the possible choices according to a search heuristic.
- Choose branch of search space randomly but biased according to a function of this ranking.
- E.g., choose branch b_i with probability:

$$\frac{bias(rank(b_i))}{\sum_j bias(rank(b_j))}$$

- Continue as in Iterative Sampling.
- Assumes a good ordering heuristic.

Carnegie Mellon

Our Approach: WHISTLING

- Motivation: heuristic more or less discriminating from context to context.
- Same basic idea as in HBSS, but decisions are biased according to a function of the heuristic value.
- E.g., choose branch b_i with probability:

$$\frac{bias(heuristic(b_i))}{\sum_j bias(heuristic(b_j))}$$

- Eliminates the O(n log n) ranking step of HBSS

Carnegie Mellon

Why “Wasp beHavior Inspired”?

- Algorithm’s name related to “how” the stochastic decision is computed
- Obvious method:
  - Pass one: compute $\sum bias(heuristic(b_j))$
  - Generate random number
  - Pass two: choose b_i with probability:

$$\frac{bias(heuristic(b_i))}{\sum bias(heuristic(b_j))}$$

- **Wasp analogy reduces this to a single pass**

Carnegie Mellon

Wasp Behavior Model [Theraulaz *et al.*, 1991]

- Each wasp of the colony has a force variable F_w
- Any two wasps may engage in a dominance contest
- Wasp 1 defeats wasp 2 with probability:

$$\frac{F_1^2}{F_1^2 + F_2^2}$$

- Winner’s force is increased; loser’s force decreased
- A social hierarchy formed over time
- **Possible analogy between most dominant wasp and most “dominant” choice?**

Carnegie Mellon

WHISTLING: Wasp beHavior Inspired STochastic sampLING

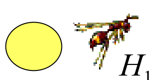
- At a decision point in the search:
  - Each choice represented by a “wasp”
  - Initial force of wasp *i*:

$$F_i = bias(heuristic(b_i))$$

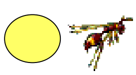
  - Tournament of dominance contests
    - Wasp *0* competes against wasp *1*
    - Winner’s force F_w accumulates loser’s force F_l
    - Loser drops out
    - Winner competes against wasp *2*, …

Illustrative Example

$$F_w = ATCS_w(t,l) = \frac{w_j}{p_j}\exp\left(-\frac{\max(d_j - p_j - t, 0)}{k_1\bar{p}}\right)\exp\left(-\frac{s_{lj}}{k_2\bar{s}}\right)$$

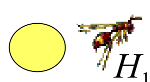


$H_1 = 0.01, F_1 = 0.0001$

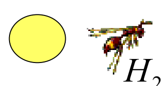


$H_2 = F_2 = 1$

$$P(W_2 \text{winning}) = \frac{1}{1.0001} = 0.999$$



$H_1 = 0.1, F_1 = 0.01$



$H_2 = 0.2, F_2 = 0.04$

$$P(W_2 \text{winning}) = \frac{0.04}{0.05} = 0.8$$

A Competing Approach

- **Heuristic-Biased Stochastic Sampling (HBSS) [Bresina, AAAI-96]**
- **Bias is based on rank ordering**

$H_2 = 1 \quad H_1 = 0.01$

$\text{rank}_2 = 1 \quad \text{rank}_1 = 2$

$$P(\text{selecting}\, J_2) = \frac{1/1^2}{1/1^2 + 1/2^2} = 0.8$$

$H_2 = 0.2 \quad H_1 = 0.1$

$\text{rank}_2 = 1 \quad \text{rank}_1 = 2$

$$P(\text{selecting}\, J_2) = \frac{1/1^2}{1/1^2 + 1/2^2} = 0.8$$

Computational Study

Experimental Design:

- Objective: Weighted tardiness
- Base heuristic: ATCS [Lee, Bhaskaran, and Pinedo 97]
- 120 problem instances
  - 60 jobs each, single machine
  - Varying degrees of due-date tightness, due-date range, and setup severity

Comparative analysis of Whistling and HBSS approaches

- Evaluation of a spectrum of bias functions for each approach
- 1, 10, and 100 restarts considered

Percentage Improvement over Deterministic ATCS Rule

	Whistling	HBSS	Whistling	HBSS	Whistling	HBSS
# Restarts	1	1	10	10	100	100
Loose due-dates	**20.29**	14.86	**45.14**	38.98	**55.35**	52.38
Medium due-dates	**2.13**	1.47	**8.38**	6.40	**13.73**	10.73
Tight due-dates	0.04	**0.21**	**0.91**	0.88	1.71	**1.83**
Severe setups	**8.12**	4.34	**20.94**	17.21	**27.03**	24.37
Moderate setups	**6.86**	6.69	**15.35**	13.63	**20.16**	18.93

Carnegie Mellon

Whistling vs Discrepancy Search

- Same problem instances as in Whistling / HBSS comparison
- Comparative analysis of Whistling, LDS, and DDS
  - 100 and 200 restarts considered for Whistling
  - LDS:
    - All single discrepancy solutions occurring in 1st four decisions (230)
    - All single discrepancy solutions (1770)
  - DDS: To depth 2 (3539)

Carnegie Mellon

Percentage Improvement over Deterministic ATCS Rule

	Whistling	LDS	Whistling	LDS	DDS
# Samples	100	230	200	1770	3539
Loose due-dates	**55.35**	52.37	**57.21**	57.14	56.75
Medium due-dates	**13.73**	11.32	**14.84**	13.63	12.18
Tight due-dates	1.71	**1.81**	**2.29**	2.12	1.83
Severe setups	**27.03**	25.08	**28.11**	26.98	26.36
Moderate setups	**20.16**	18.59	21.45	**21.61**	20.82

CPU Time

HBSS	HBSS	Whistling	Whistling	Whistling	LDS	LDS	DDS
10	100	10	100	200	230	1770	3539
1.59 s	15.46	0.16 s	1.50 s	3.01 s	1.46	6.04	20.94

- Note:
  - 100 iterations of Whistling in same time as 10 iterations of HBSS
  - 100 iterations of Whistling in same time as considering all 230 single discrepancy solutions in first 4 decisions
  - 200 iterations of Whistling in half the time of considering all 1770 single discrepancy solutions
  - 200 iterations of Whistling in a seventh of the time to consider the 3539 solutions of a DDS to depth 2

References

[Bresina 96] J. Bresina. Heuristic-biased stochastic sampling. In *Proc. 13th Nat. Conf. AI*, 1996.

[Cesta, Oddi, & Smith 99] A. Cesta, A. Oddi, and S. Smith. An iterative sampling procedure for resource constrained project scheduling with time windows. In *Proc. 16th IJCAI*, 1999.

[Gomes, Selman, & Kautz 98] C. Gomes, B. Selman, and H. Kautz. Boosting combinatorial search through randomization. In *Proc. 15th Nat. Conf. AI*, 1998.

[Harvey & Ginsberg 95] W. Harvey and M. Ginsberg. Limited discrepancy search. In *Proc. 14th IJCAI*, 1995.

[Langley 92] P. Langley. Systematic and nonsystematic search strategies. In *Proc. 1st AIPS*, 1992.

[Oddi & Smith 97] A. Oddi and S. Smith. Stochastic procedures for generating feasible schedules. In *Proc. 14th Nat. Conf. AI*, 1997.

[Theraulaz et al. 91] G. Theraulaz, S. Goss, J. Gervet, and J. Deneubourg. Task differentiation in polistes wasp colonies: a model for self-organizing groups of robots. In *Proc. 1st Int. Conf. On Simulation of Adaptive Behavior*, 1991.

[Walsh 97] T. Walsh. Depth-bounded discrepancy search. In *Proc. 15th IJCAI*, 1997.