Université Libre de Bruxelles

*Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle*

IRIDIA

A Stochastic Routing Policy Perspective for an Operator-Level Hyperheuristic

Christophe Philemotte

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2008-018

August 2008

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:

IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2008-018

A Stochastic Routing Policy Perspective for an Operator-Level Hyperheuristic

Christophe Philemotte cphilemo@iridia.ulb.ac.be
http://iridia.ulb.ac.be/~cphilemo
IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium

12 August 2008

Abstract

The search process of a metaheuristic can be seen as one agent or several agents that walk on the neighborhood graph G associated with the tackled problem instance π. Graph nodes are candidate solutions. Optimizing π then consists in routing agents, in a minimal amount of moves, to an unknown goal, i.e. the global optimum of π. The routing policy corresponds to the operators the metaheuristic uses. Typically, they are stochastic and define the transition probability between the visited node and their neighbors. These probabilities are tuned off-line by a few parameters. The policy is then ruggedly fixed. This paper addresses the problem of adapting the policy, i.e. the operators, during the search. This online learning aims at improving the routing through the search space, in other words minimizing the computation time to go to the optimum. The paper concludes on a hyperheuristic framework that considers the online adaptation of operators. **keywords:** Metaheuristic, Reinforcement Learning, Hyperheuristic, Operators

1 Preliminaries

A Combinatorial Optimization Problem (COP) could be defined as a set of variables $X := \{x_1, \ldots, x_n\}$, their respective domains $\mathbb{D}_i, \forall i = 1, \ldots, n$, possible constraints $c_j\left(Y_j\right), Y_j \subseteq X, \forall j = 1, \ldots, m$ and an objective function $f : \bigotimes_{i=1}^{n} \mathbb{D}_i \to \mathbb{R}$ to be optimized.

A metaheuristic can only operate in the search space $S(\pi) = \bigotimes_i \mathbb{D}_i$ of problem instance π with respect of the neighborhood relation $N(\pi)$. This relation $N(\pi)$ depends partially on the chosen representation that encodes the problem. As Hoos and Stüzle stated [4], the search process of a metaheuristic can be seen as a walk on the neighborhood graph associated with instance π, $G(\pi) := (S(\pi), N(\pi))$. This neighborhood graph describes the possible search actions, i.e. operators. Metaheuristics have to respect this structure and cannot operate out of. The neighborhood relation $N(\pi)$ is specified by the function $N : S(\pi) \to 2^{S(\pi)} : s \to N(s) := \{s' \in S | N(s, s')\} \subseteq S$, where s is a candidate solution and the set $N(s)$ is called the neighborhood of s.

A stochastic operator is a function that applies s on a $s' \in N(s)$ with a given transition probability $p_{ss'} = P(s'|s)$. All these probabilities are not given explicitly. They are typically induced by other probabilities that are used in the operator algorithm and tuned by one or more parameter. Sometimes, they are governed by some measures such as the evaluation of the candidate solution. For easy instance, in Simple Genetic Algorithm (SGA), the bit flip mutation flips each bit of the bit string, i.e. the chromosome, with a probability that is given as a rate parameter $\mu \in [0, 1]$. If the chromosome is l bits long, and the hamming distance between s and s' is $m \in [0, l]$, the probability $p_{ss'}$ is equal to $\mu^m (1 - \mu)^{l-m}$.

2 A Stochastic Routing Perspective

The metaheuristic process is usually described as iterations of operators. The process is iterated until some termination conditions are met. They could be based on time consumption or convergence criterion or other measures. For trajectory based metaheuristics, the process is trivially like an agent walking on the neighborhood graph G. For population based metaheuristics, we can consider a population of agents that can eventually communicate together. In both cases, one or more agents walk on G and search for the global optimum. By definition of optimization, this optimum is not known. The agent does not know if the visited node is the goal. Each agent step is defined by operators and cost a same amount of time: one iteration time unit. An agent could be rewarded positively or negatively. The reward $r_{ss'}$ depends on the agent progress to the goal. This scenario can be considered as a stochastic routing of agents [1]. The routing policy is nothing else than the tuned operators that the metaheuristic uses. The optimal routing policy is the set $\varpi = \{p_{ss'}\}$ that maximizes the total reward R_{ϖ}, i.e. the measure of the closeness to the goal.

We can now ask the following question: "For a given problem instance π and a given amount of computation time, how the optimal policy ϖ could be learned?". This question states the problem as a Reinforcement Learning (RL) of a Partially Observable Markov Decision Process (POMDP). Due to the size of COP search space $|S(\pi)|$, it is very computation intensive to apply value based method to compute iteratively the optimal ϖ. In this case, a more adequate RL method would be searching directly in the policy space [7]. As explained before, we remember that a specific operator implies some relations between the transition probabilities $p_{ss'}$. These relations are constraints on the optimization of ϖ.

3 An Operator-Level Hyperheuristic

In previous section, we have defined a specific perspective where operators are considered as policies. This perspective gives foundations for new hyperheuritics that adapt operators online. There are several possibilities to resolve the previous stated RL problem: tuning operator parameters, redefining or reprogramming the operator, or biasing the induced $p_{ss'}$. For both first and second, there already exist some implementation such as relaxation of temperature in simulated annealing [5], adaptation of mutation rate in evolutionary algorithm [3], or genetic programming that evolves operator algorithm [6]. Here, we will give more attention to the last one: biasing the induced $p_{ss'}$.

One way of biasing transition probabilities is by altering the graph G. We remember that the graph G mainly depends on the problem representation. If the representation is adapted, the graph G is altered, and the $p_{ss'}$ is biased. Indeed, a new representation can delete or create some neighborhood transition. The effect is nothing or all. It is however a very meaning way to guide the search process, especially in our routing perspective. There exist some works that only make permutations in the representation [2]. The permutations affect principally the distance between two neighbors. One other possible alteration of the representation consists in the merge of variables to make a new one. This approach can deeply modify the graph G, as by cutting in two unconnected subgraphs. For instance, it is the approach used by the gestalt hyperheuristic introduced in [8].

4 Conclusion

By building a bridge between optimization and stochastic routing problem, we have stated foundations for a new class of hyperheuristics addressed to the online adaptation of operators. We have discussed about possible implementation approach especially one that alters the encoding such as the gestalt hyperheuristic.

References

[1] Y. Achbany, F. Fouss, L. Yen, A. Pirotte, and M. Saerens. Tuning continual exploration in reinforcement learning. In S. Kollias, A. Stafylopatis, W. Duch, and E. Oja, editors, *Proceedings of the 16th International Conference on Artificial Neural Networks (ICANN 06)*, volume 4132 of *Lecture notes in Computer Science*. Springer, September 2006.

[2] L. Barbulescu, J.-P. Watson, and D. Whitley. Dynamic representations and escaping local optima: Improving genetic algorithms and local search. In *AAAI/IAAI*, pages 879–884, 2000.

[3] T. Bäck. An overview of parameter control methods by self-adaption in evolutionary algorithms. *Fundamental Informaticae*, 35(1-4):51–66, 1998.

[4] H. H. Hoos and T. Stützle. *Stochastic Local Search : Foundations & Applications*. Elsevier, 2004.

[5] L. Ingber. Adaptive simulated annealing (asa): Lessons learned. *Control And Cybernetics and Systems : an International Journal*, 25(1):33–54, 1996.

[6] W. Kantschik, P. Dittrich, M. Brameier, and W. Banzhaf. Meta-evolution in graph gp. In *Second European Workshop, EuroGP'99 Göteborg, Sweden, May 26–27, 1999*, volume 1598/1999. Springer Berlin / Heidelberg, 1999.

[7] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette. Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research*, 11:241–276, 1999.

[8] C. Philemotte and H. Bersini. A gestalt genetic algorithm: less details for better search. In H. Lipson, editor, *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1328–1334, New York, NY, USA, 2007. ACM.