# Improving Search Algorithms by Using Intelligent Coordinates

David Wolpert[1], Kagan Tumer[1], and Esfandiar Bandari[2]
[1]*NASA and* [2]*USRA/RIACS, Ames Research Center, Moffett Field, CA 94035,* {dhw,kagan,bandari}@email.arc.nasa.gov

We consider algorithms that maximize a global function $G$ in a distributed manner, using a different adaptive computational agent to set each variable of the underlying space. Each agent $\eta$ is self-interested; it sets its variable to maximize its own function, $g_\eta$. Three factors govern such a distributed algorithm's performanc, related to exploration-exploitation, game theory, and machine learning. We demonstrate how to exploit all three factors by modifying a search algorithm's exploration stage: rather than random exploration, each coordinate of the search space is now controlled by a separate machine-learning-based "player" engaged in a non-cooperative game. Experiments demonstrate that this modification improves Simulated Annealing (SA) by up to an order of magnitude for bin-packing and for a model of an economic process run over an underlying network. These experiments also reveal novel small worlds phenomena.

PACS numbers: 89.20.Ff,89.75.-k ,89.75.Fb,02.60.Pn, 02.70.-c, 02.70.Tt

## I. INTRODUCTION

Many systems found in nature have inspired computational algorithms for how to maximize a provided high-dimensional function. In some of these algorithms the values of the underlying coordinates are controlled by separate players engaged in a non-cooperative game; their equilibrium joint state (hopefully) maximizes the provided global function $G$. Examples of such systems are auctions and clearing of markets. Typically, in the computational algorithms inspired by such "collectives" of players, each "player" is instantiated as a separate machine learning algorithm [3, 6], e.g., a reinforcement-learning (RL) algorithm [10, 14].

There are three crucial issues concerning such collectives. The first is whether the payoff function $g_\eta$ of each player $\eta$ is sufficiently sensitive to the coordinate $\eta$ controls in comparison to the other coordinates. If this is not the case, it is not feasible for $\eta$ to learn how to set its coordinate to achieve high payoff. The second crucial issue is the need for all of the $g_\eta$ to be "aligned" with $G$, so that as the players individually learn how to increase their payoffs, $G$ also increases.

Other collective systems found in nature that have inspired function-maximization algorithms do not involve players conducting a non-cooperative game. Examples include equilibrating spin glasses, genomes undergoing neo-Darwinian natural selection, and eusocial insect colonies. These have been translated into simulated annealing (SA [7]), genetic algorithms [1], and swarm intelligence [2], respectively. The third crucial issue is most prominent in such algorithms: the need to tradeoff exploration and exploitation.

Recent analysis [12] reveals that a collective's $G$ value is governed by the interaction between these three effects: the alignment of the $g_\eta$ with $G$, the "learnability" of the $g_\eta$, and the exploration/exploitation tradeoff [13]. These three issues are traditionally studied in three separate fields: game theory, machine learning/statistics, and optimization theory, respectively. So any complete science of collectives must incorporate insights from all three fields; no single one of the fields suffices.

Previous work in the COllective INtelligence (COIN) framework has partially accomplished this, addressing the first two issues. That work is an extension of game-theoretic mechanism design, to include off-equilibrium behavior, learnability issues, non-human $g_\eta$ (e.g., $g_\eta$ for which incentive compatibility is irrelevant), and arbitrary $G$ [5, 9]. In domains from network routing to congestion problems COIN-based algorithms beat traditional techniques, by up to several orders of magnitude [13, 14].

Here we address all three issues at once, by replacing the exploration step in any exploration/exploitation search algorithm. In the new exploration step each separate coordinate of the multi-coordinate space being searched is made "intelligent", its value being the move of a player/agent designed using the COIN framework (rather than the value of a random sample of a proposal distribution). We call the class of such algorithms Intelligent Coordinates for search (IC).

We concentrate on IC with SA as the exploration-based search algorithm. Like SA, IC is intended to be used "off the shelf"; rarely will it be the best possible algorithm for a particular domain. Also like SA, IC is best suited to very large problems (so parallelization can be exploited), where there is little exploitable gradient information.

We present experiments comparing IC and SA on two archetypal domains: bin-packing and an economic model of people choosing formats for their home music systems. In bin-packing IC achieves a given value of $G$ up to three orders of magnitude faster than does SA, an improvement ratio that increases linearly with problem size. In the format choice problem, each person $\eta$'s move is the choice of which several formats to adopt, and $G$ is the sum of everyone's "happiness" with their move. In turn, $\eta$'s happiness with each of the formats making up her move is set by three factors: which of her nearest neighbors on a ring network ($\eta$'s "friends") choose that format; $\eta$'s intrinsic preference for that format; and the price of music purchased in that format, inversely proportional to the total number of players using that choice. Here IC improves $G$ two orders of magnitude faster than SA. We also tested an algorithm designed to "endogenize externalities", in the language of economics; IC outperformed it by over two orders of magnitude. We also replaced the ring with a small-worlds network [11]. This barely improved IC's performance (3%), and had no effect on the other algorithms. However if $G$ too was changed, so that $\eta$'s happiness depends on agreeing with her friends' friends, the improvement is significant (10%).

## II. SIMPLIFIED THEORY OF COLLECTIVES

Let $z \in \zeta$ be the joint move of all agents/players in the collective, with agent $\eta$'s move being $z_\eta$, and $z_{-\eta}$ being the other agents' moves. We wish to find the $z$ maximizing the provided **world utility**, $G(z)$. We also have **private utilities** $\{g_\eta\}$, one for each agent $\eta$. These are the functions that the individual agents try to maximize.

It will be useful to "standardize" utility functions so that the value they assign to $z$ only reflects their ranking of $z$ relative to other possible points in $\zeta$. Such a standardization is called **intelligence**, one form of which is

$$N_{\eta,U}(z) \equiv \int d\mu_{z_{-\eta}}(z')\Theta[U(z) - U(z')] \ , \qquad (1)$$

where $\Theta$ is the Heaviside function, and where the subscript on the (normalized) measure $d\mu$ indicates it is restricted to $z'$ such that $z'_{-\eta} = z_{-\eta}$. Intuitively, $N_{\eta,U}(z) \in [0.0, 1.0]$ is the percentile rank of $\eta$'s choice of move in comparison to her alternatives. The ranking is according to utility $U$, and is in the context of the other agents making move $z_{-\eta}$.

We define $\vec{N}_G$ and $\vec{N}_g$ as the vectors of the intelligences of all agents, for the world utility and the agents' separate private utilities, respectively. $N_{\eta,g_\eta}(z) = 1$ means that agent $\eta$'s move maximizes its utility, given the moves of the other agents. So in game theory terms, $\vec{N}_g(z) = \vec{1}$ means $z$ is a Nash equilibrium. Conversely, $\vec{N}_G(z') = \vec{1}$ means that the value of $G$ cannot increase in moving from $z'$ along any single coordinate of $\zeta$.

Indicate the agents' private utilities by $s$. Our uncertainty about the system induces a distribution $P(z)$. Bayes' theorem gives us the associated $P(G \mid s)$:

$$\int d\vec{N}_G P(G \mid \vec{N}_G, s) \int d\vec{N}_g P(\vec{N}_G \mid \vec{N}_g, s) P(\vec{N}_g \mid s) \quad (2)$$

This is the **central equation**. Say that for some $s$ the third conditional probability in the integrand is peaked near $\vec{N}_g = \vec{1}$, i.e., $s$ probably induces large (private utility) intelligences. If in addition the second probability term is peaked near $\vec{N}_G = \vec{N}_g$, then $\vec{N}_G$ is also large. In particular, if $s$ guarantees that $\vec{N}_g$ equals $\vec{N}_G$ identically $\forall z$, then the second term is a delta function, regardless of $P(z \mid s)$. Such a system is called **factored**. Finally, say that in addition to such second and third terms, the first term is peaked about high $G$ whenever $\vec{N}_G$ is large. Then as desired, $s$ (probably) induces high $G$.

The second term is related to game theory. As an example, a **team game**, where $g_\eta = G \ \forall\eta$, is factored [13]. However team games usually have poor third terms, especially in large collectives. This is because agent $\eta$ chooses its move based on what it can learn about the effect of that move on $g_\eta$. This learning is based on previous observations of what value $g_\eta$ had when $\eta$ made its various moves. These observations are necessarily made in the presence of the (varying) moves of all the other agents. So say $g_\eta = G$, and the moves of the other agents affect $G$ comparably to how much $\eta$'s move does. Then when there are many of those other agents, it will be difficult to distinguish the "signal", of the effect of $\eta$'s move on the value of $g_\eta$, from the "noise", of the effect of other agents' moves. This will make it difficult for $\eta$ to learn how to make a move that has high intelligence for $g_\eta$.

Term three is related to machine learning. Say that we IID sample the values that the utility $g_\eta$ has whenever the move made by agent $\eta$ is $z^1_\eta$, and similarly for when that move is $z^2_\eta$. This gives us a **training set** of move-utility pairs, $n_\eta$. The associated **learnability**, $\Lambda(U; n_\eta, {z_\eta}^1, {z_\eta}^2)$ is defined as

$$\frac{|E(g_\eta({z_\eta}^1, .) \mid n_\eta) - E(g_\eta({z_\eta}^2, .) \mid n_\eta)|}{\sqrt{Var(g_\eta(z^1_\eta, .) \mid n_\eta) + Var(g_\eta(z^2_\eta, .) \mid n_\eta)}} \ , \qquad (3)$$

where the expectations and variances have $\eta$'s move fixed as indicated, with $z_{-\eta}$ varying according to $P(z_{-\eta} \mid n_\eta)$.

The denominator in Eq. 3 reflects the sensitivity of $g_\eta(z)$ to $z_{-\eta}$, while the numerator reflects its sensitivity to $z_\eta$. So the greater $\Lambda(g_\eta; n_\eta, {z_\eta}^1, {z_\eta}^2)$, the more $g_\eta(z)$ depends on which of those two moves agent $\eta$ adopts, in comparison to its dependence on the joint move of the other agents. In other words, for larger learnability it is easier for $\eta$ to distinguish in $n_\eta$ how its choice between those two moves affects $g_\eta$ (the signal), from how other agents affect $g_\eta$ (the noise). Formally, the expected intelligence of agent $\eta$'s move is an increasing function of the value of $\Lambda(g_\eta; n_\eta, {z_\eta}^1, {z_\eta}^2)$ for all candidate pairs of moves, ${z_\eta}^1, {z_\eta}^2$. So if $s$ specifies a $g_\eta$ with high learnability $n_\eta$, term 3 will have the desired form.

A **difference** utility has the form $g_\eta(z) = G(z) - D(z_{-\eta})$. Any such utility is factored [12]. The $D(z_{-\eta})$ that maximizes $\Lambda(g_\eta; n_\eta, {z_\eta}^1, {z_\eta}^2)$, for all pairs ${z_\eta}^1, {z_\eta}^2$, is $\int dz_\eta f(z_\eta) G(z_\eta, z_{-\eta})$ [12] (the form of the distribution $f$ is beyond the scope of this paper). The associated $g_\eta$ is called the **Aristocrat** utility ($AU$). If $\eta$'s private utility is $AU$ rather than $G$, then the last two terms of the central equation are more biased towards higher world utility values. Another advantage is that $AU$ is often easier to evaluate than is $G$[13].

The **Wonderful Life** Utility (WLU) is an approximation of $AU$ that avoids calculating expectation values by adopting delta function $f$:

$$WLU_\eta \equiv G(z) - G(z_{-\eta}, CL_\eta) \ , \qquad (4)$$

where $CL_\eta$ is the **clamping parameter**. The choice of $CL_\eta$ can be set to maximize learnability. (How best to do this is beyond the scope of this paper; see below.) While not matching that of $AU$, for most $CL_\eta$ values $WLU$'s learnability is better than a team game's.

Finally, one way to address term 1 of the central equation is to incorporate exploration/exploitation techniques like SA. We describe how to do this below.

## III. EXPERIMENTS

In our version of SA, at the beginning of each time-step $t$ a proposal distribution $h_\eta(z_\eta)$ is formed for every separate coordinate $\eta$. Each such distribution assigns probability 75% to the move $\eta$ had at the end of the preceding time-step, $z_{\eta,t-1}$, and uniformly divides probability 25% across the other moves. The "exploration" joint-move $z_{expl}$ is then formed by simultaneously sampling all the $h_\eta$. If $G(z_{expl}) > G(z_{t-1})$, $z_{\eta,t}$ is set to $z_{expl}$. Otherwise $z_t$ is set by sampling a Boltzmann distribution over the two $z$'s for "energies" $G(z_{t-1})$ and $G(z_{expl})$, respectively. Many different annealing schedules were investigated; all results below are for best schedules found.

IC is identical except that each $h_\eta$ is replaced by $\frac{h_\eta(z_\eta)c_{\eta,t}(z_\eta)}{\sum_{z'_\eta} h_\eta(z'_\eta)c_{\eta,t}(z'_\eta)}$. To form $c_{\eta,t}$, for each of its possible moves $z_\eta$, agent $\eta$ collects those instances in its training set for which the move was $z_\eta$. It then forms a weighted average of the associated $g_\eta$ values recoreded in the training set. (The weights decay exponentially with how old that pair is, to reflect nonstationarity of the system.) This gives an estimate of what $g_\eta$ is likely to be for each move $z_\eta$. $c_{\eta,t}$ then is the Boltzmann distribution over the possible $z_\eta$, parameterized by a "learning temperature", with the "energy" for each $z_\eta$ set to the associated estimate of what $g_\eta$ is likely to be. ("COIN" algorithms just use this Boltzmann distribution to give $z_t$ directly, with no proposal distribution, keep/reject step, etc.)

In all our experiments "AU" used a mean-field approximation to pull the expectation inside the $G(.)$ in the evaluation of $D(z_{-\eta})$. Unless otherwise specified, for simplicity, $CL_\eta$ (used in WLU) was set to $\vec{0}$.

| Algorithm | Ave. $G$ | Best | Worst | % Optimum |
|---|---|---|---|---|
| IC WLU | $3.32 \pm 0.22$ | 2 | 8 | 72 % |
| IC TG | $7.84 \pm 0.17$ | 6 | 10 | 0 % |
| COIN WLU | $3.52 \pm 0.20$ | 2 | 7 | 64 % |
| COIN TG | $7.84 \pm 0.15$ | 6 | 9 | 0 % |
| SA | $6.00 \pm 0.19$ | 4 | 7 | 0 % |

TABLE I: Bin-packing $G$ at time 200 for $N = 20, c = 12$.

In the bin-packing problem, $N$ items, all of size $< c$, must be assigned into a minimal subset of $N$ bins, without assigning a summed size $> c$ to any one bin. $G$ of an assignment pattern is the number of occupied bins [4], and each agent controls the bin choice of one item. All algorithms use a modified "$G$", $G_{\text{soft}}$, even though their performance is ultimately measured with $G$:

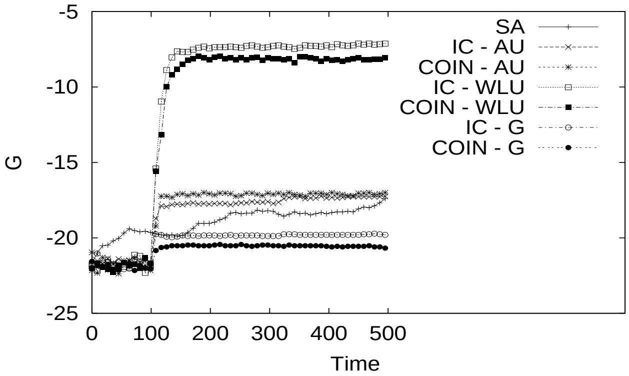


FIG. 1: Average bin-packing $G$ for $N = 50, c = 10$. All error bars $\leq .31$ except IC - AU and COIN - AU are $\leq .57$.

$$G_{\text{soft}} \equiv \begin{cases} \sum_{i=1}^{N} \left[ \left(\frac{c}{2}\right)^2 - \left(x_i - \frac{c}{2}\right)^2 \right] & \text{if } x_i \leq c \\ \sum_{i=1}^{N} \left(x_i - \frac{c}{2}\right)^2 & \text{if } x_i > c \end{cases}, \tag{5}$$

where $x_i$ is the summed size of all items in bin $i$.

In the IC runs learning temperature was .2, and all agents made the transition to RL-based moves after a period of 100 purely random $z$'s that was used to generate the initial training sets $\{n_\eta\}$. Exploitation temperature started at .5 for all algorithms, and was multiplied by .8 every 100 exploitation time-steps. In each SA run, $h$ was slowly modified to generate solutions that differed less from the current solution as time progressed.

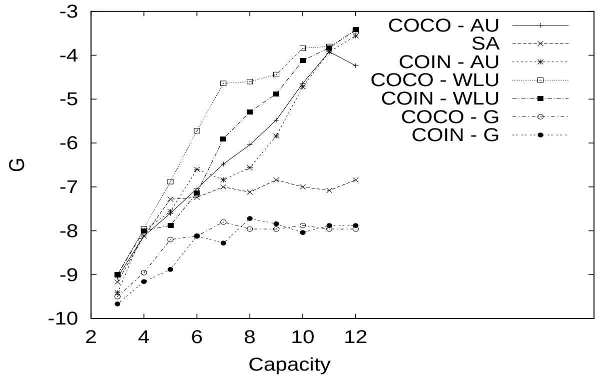


FIG. 2: $G$ vs. $c$ for $N = 20$ at $t = 200$. All error bars $\leq .34$.

In Table 1 "Best" and "worst" give the extremal end-of-the-run $G$ values (25 runs total), and "%Optimum" the percentage of runs within one bin of the best value. Fig. 1 illustrates that the algorithms that account for both terms 2 and 3 — IC WLU and COIN WLU — far outperform the others, with the algorithm accounting for all three terms doing best. The worst algorithms were those that accounted for only a single term (SA and COIN TG). Linearly (i.e., optimistically) extrapolating SA's performance from time 15000 indicates it would take over 1000 times as long as IC WLU to reach the $G$ value IC WLU reaches at time 200. In addition the ratio of WLU's time 1000 performance (relative to random search) to SA's grows linearly with the size of the problem. Finally, Fig. 2 illustrates that the benefit of addressing terms 2 and 3 grows with the difficulty of the problem. In both figures SA outperforms IC - TG due to its benefiting from more parameter-tuning.

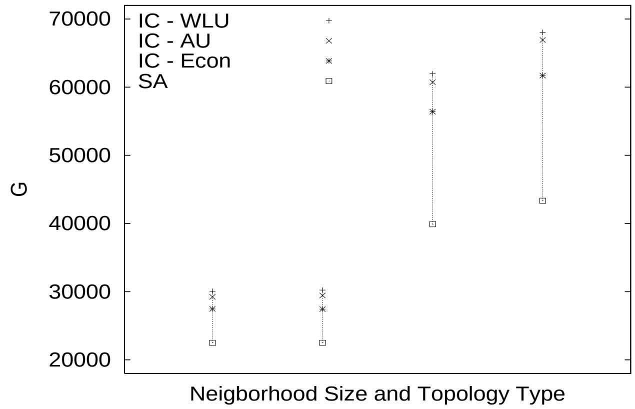


FIG. 3: $G(t = 200)$ for 100 agents. In order from left to right, $D = \{1, 1, 3, 3\}$, and topologies are {L,W,L,W}.

For the format choice problem $G$ is the sum over all $N_a$ agents $\eta$ of $\eta$'s "happiness" with its music formats: $G = \sum_{\eta=1}^{N_a} \sum_{i=1}^{N_f} \sum_{\eta' \in neigh_\eta} \vartheta(i) \; \omega_{\eta,\eta',i} \; pref_{\eta,i}$, where $N_f$ is the numbers of formats; $neigh_\eta$ is the set of players $\leq D$ hops away from player $\eta$; $pref_{\eta,i}$ is $\eta$'s intrinsic preference for format $i$ (randomly fixed $\in [0, 1]$); $\vartheta_(i)$ is the total number of players choosing format $i$ (i.e., the inverse price for format $i$); and $\omega_{i,\eta,\eta'} = 1$ if the choices of players $\eta$ and $\eta'$ both include format $i$, 0 otherwise. $\eta$'s move says which of four formats *not* to use. For example, $WLU_\eta = \sum_{\eta,i,\eta' \in neigh_\eta} \vartheta(i) \; \Omega_{\eta',i} \; pref_{\eta,i}$, where $\Omega_{\eta',i} = -1$ if agent $\eta'$'s choices include $i$ but agent $\eta$'s do not, and equals 0 otherwise . Both $D = 1$ and 3 were investigated.

In Fig. 3, "IC Econ" refers to WLU IC, with clamping making the agent decline all formats. It is a crude way of endogenizing externalities, and then rescaling and interleaving with SA to improve performance. "IC-WLU" instead clamps $\eta$'s move to zero (as per the theory of collectives), so that $\eta$ chooses all formats. Learning temperature was .4, and exploitation temperature was .05 (annealing provided no advantage since runs were short). We used $m$-node ring topologies with an extra $.06m$ random links added, a new such set for each of the 50 runs giving a plotted average value. "Short links" (L) means all extra links connected players two hops apart, and "small-worlds" (W) means there was no such restriction.

IC Econ's inferior performance illustrates the shortcoming of economics-like algorithms. For $D = 1$ SA did not benefit from small worlds connections, and IC variants barely benefited ( 3%), despite the associated drop in average inter-node hop distance. However if $D$ also increased, so that $G$ directly reflected the change in the topology, then the gain with a small worlds topology grew to 10%.

The authors thank Michael New, Bill Macready, and Charlie Strauss for helpful comments.

[1] T. Back, D. B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. Oxford University Press, 1997.

[2] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Nature*, 406(6791):39–42, 2000.

[3] G. Caldarelli, M. Marsili, and Y. C. Zhang. *Europhysics Letters*, 40:479–484, 1997.

[4] E. G. Coffman Jr., G. Galambos, S. Martello, and D. Vigo. In *Handbook of Combinatorial Optimization*. Kluwer Academic Publishers, 1998.

[5] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.

[6] B. A. Huberman and T. Hogg. In *The Ecology of Computation*, pages 77–115. North-Holland, 1988.

[7] S. Kirkpatrick, C. D. Jr Gelatt, and M. P. Vecchi. *Science*, 220:671–680, May 1983.

[8] M.J.B. Krieger, J.-B. Billeter, and L. Keller. *Nature*, 406:992–995, 2000.

[9] N. Nisan and A. Ronen. *Games and Economic Behavior*, 35:166–196, 2001.

[10] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[11] D. J. Watts and S. H. Strogatz. *Nature*, 393:440–442, 1998.

[12] D. H. Wolpert. Theory of collective intelligence. In K. Tumer and D. H. Wolpert, editors, *Collectives and the Design of Complex Systems*, New York, 2003. Springer.

[13] D. H. Wolpert and K. Tumer. *Advances in Complex Systems*, 4(2/3):265–279, 2001.

[14] D. H. Wolpert, K. Wheeler, and K. Tumer. *Europhysics Letters*, 49(6), March 2000.