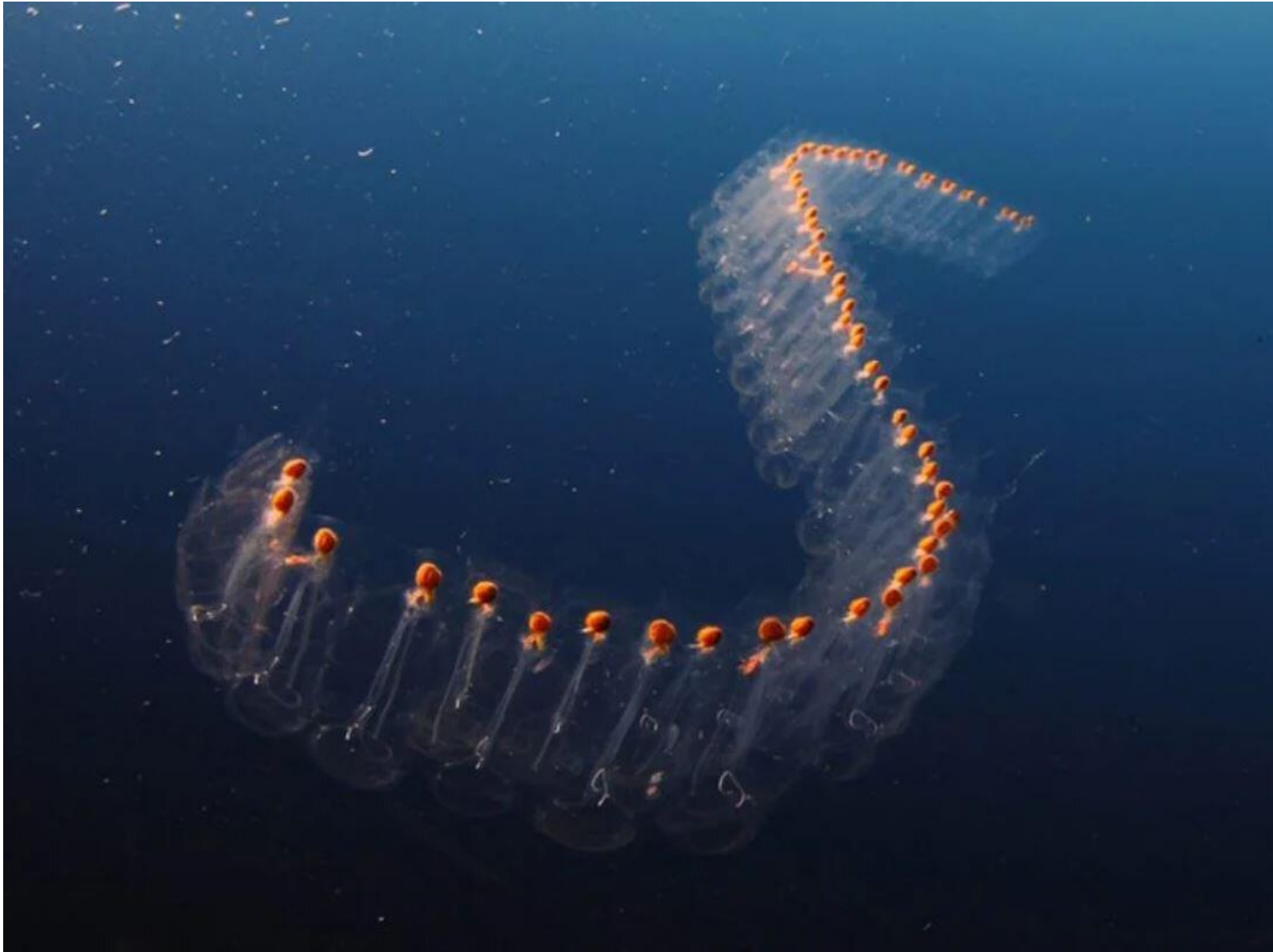


Оптимизация методом сальп,
алгоритмом стаи криля,
алгоритмом кузнечика

SALP SWAM ALGHORITM (SSA)



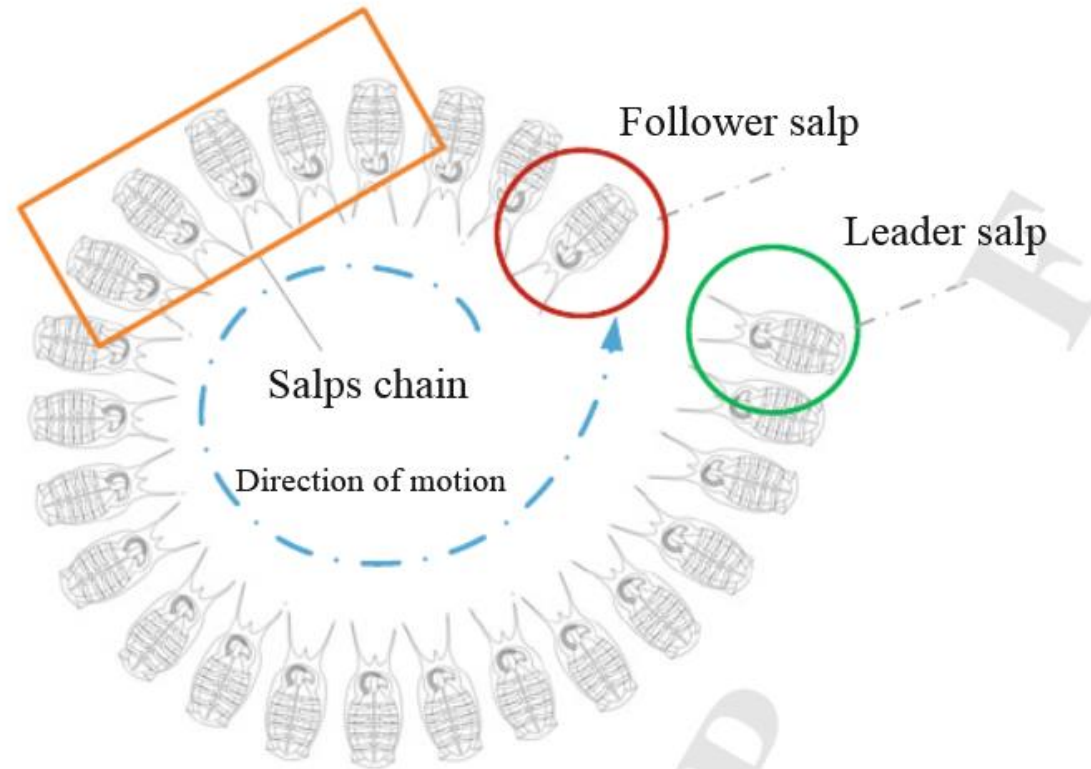


Fig. 2 Illustration of salp's chain and the concept of leader and follower

Модель алгоритма

- 1) Инициализация популяции
- 2) Движение лидера
- 3) Движение ведомых сальп

1. Инициализировать популяцию сальп
2. Вычислить приспособленность каждой особи
3. Найти лучшую сальпу (лидера)
4. Пока не достигнут критерий остановки:
 5. Обновить коэффициент c_1
 6. Для каждой сальпы (кроме лидера):
 7. Если это лидер:
 8. Обновить позицию по формуле лидера
 9. Иначе:
 10. Обновить позицию по формуле ведомых
 11. Проверить границы поиска
 12. Обновить лучшую сальпу
13. Вернуть лучшее решение

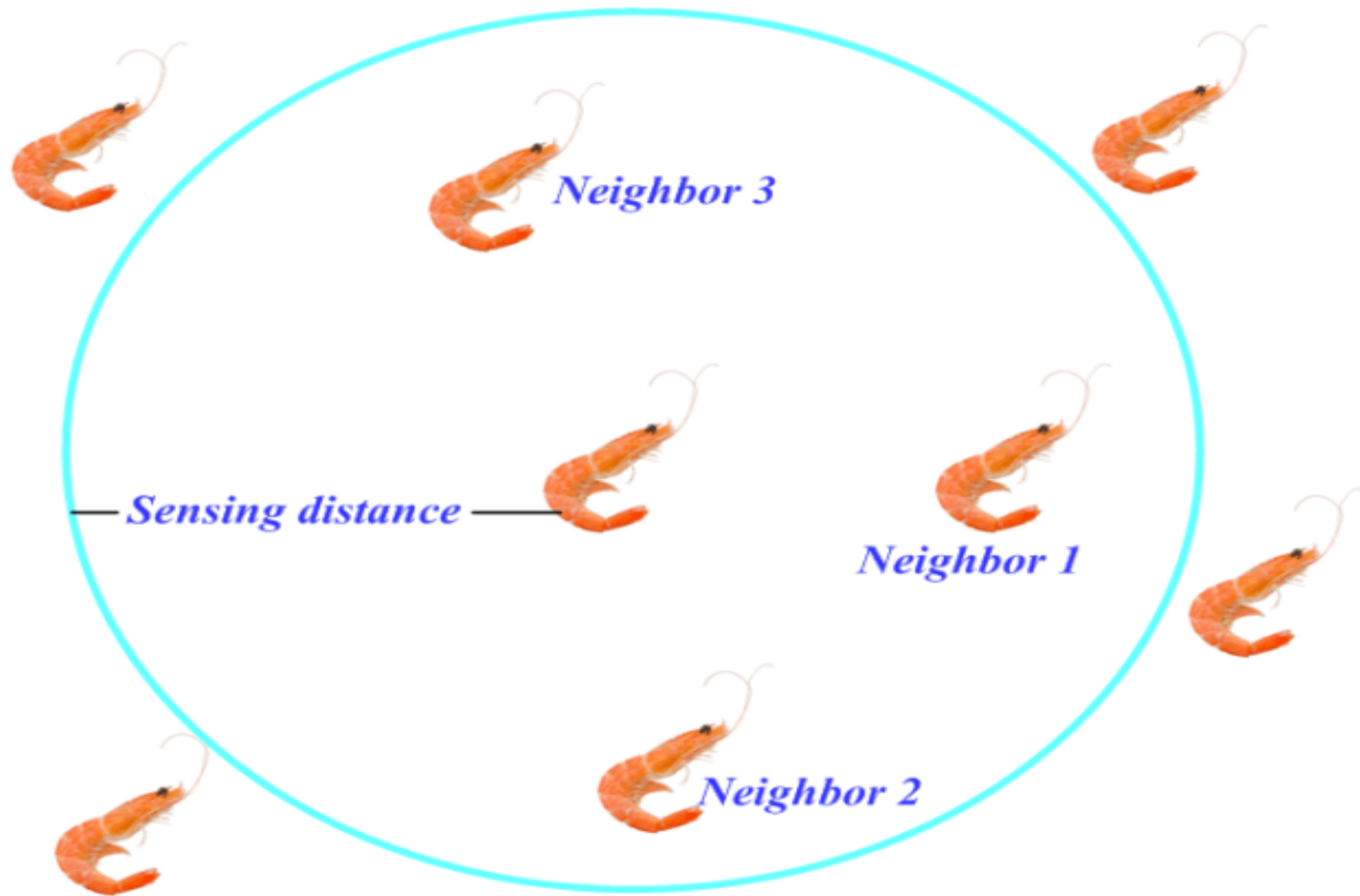
$$c_1 = 2 \cdot e^{-(4 \cdot t / T)^2}$$

$$x_1^j = \begin{cases} F_j + c_1 \cdot ((ub_j - lb_j) \cdot c_2 + lb_j), & \text{если } c_3 \geq 0.5 \\ F_j - c_1 \cdot ((ub_j - lb_j) \cdot c_2 + lb_j), & \text{если } c_3 < 0.5 \end{cases}$$

$$x_i^j = \frac{1}{2}(x_i^j + x_{i-1}^j)$$

KRILL HERD ALGORITHM (KHA)





Модель алгоритма

- 1) Движение к пище
- 2) Взаимодействие с соседями
- 3) Случайное блуждание

1. Инициализировать популяцию криля
2. Вычислить `fitness` для каждой особи
3. Найти лучшую позицию (`X_food`)
4. Для каждой итерации `t`:
 5. Для каждой особи `i`:
 6. Обновить `N_i` (взаимодействие с соседями)
 7. Обновить `F_i` (движение к пище)
 8. Обновить `D_i` (случайное блуждание)
 9. Обновить позицию: $X_i += dX_i/dt$
 10. Проверить границы
 11. Обновить `X_food`
12. Вернуть лучшее решение

$$\frac{dX_i}{dt} = N_i + F_i + D_i$$

$$N_i^{\text{new}} = N^{\text{max}} \cdot \left(\sum_{j=1}^K \frac{X_j - X_i}{\|X_j - X_i\|} \right) + \omega_n \cdot N_i^{\text{old}}$$

$$F_i = C^{\text{food}} \cdot (X^{\text{food}} - X_i) + \omega_f \cdot F_i^{\text{old}}$$

$$D_i = D^{\text{max}} \cdot \left(1 - \frac{t}{T} \right) \cdot \delta$$