Combining Stellar Wind Algorithms with Quantum Circuit Algorithms

Stellar Wind Algorithms

1. Plasma Wave (Langmuir) Frequency

```
def plasma_frequency(omega_L, k, v_T):
    return (omega_L**2 + 3 * k**2 * v_T**2)**0.5
```

2. Velocity Distribution of Energetic Electrons (Loss Cone Function)

```
import math

def loss_cone_distribution(v_parallel, v_perp, n_s, m_e, k_B, T_s):
    factor = n_s / (4 * math.pi * (2 * math.pi)**1.5 * (m_e * k_B * T_s)**2.5)
    exp_part = math.exp(-m_e * (v_parallel**2 + v_perp**2) / (2 * k_B * T_s))
    return factor * v_perp**2 * exp_part
```

3. Growth Rate of the Loss Cone Instability

```
def growth_rate(omega_p, k, v_Ts, n_s, n, alpha):
    cot_alpha = 1 / math.tan(alpha)
    factor1 = 1 / 4 * (math.pi / 2)**0.5 * omega_p**4 / (k**3 * v_Ts**3) * n_s / n
    factor2 = (omega_p**2 / (k**2 * v_Ts**2) + 2 * cot_alpha**2 - 1) / (1 + cot_alpha**2)
    exp_part = math.exp(-omega_p**2 / (2 * k**2 * v_Ts**2))
    return factor1 * factor2 * exp_part
```

4. Unstable Phase Velocities

```
def is_unstable(v_ph, v_Ts, alpha):
    cot_alpha = 1 / math.tan(alpha)
    return v_ph**2 < v_Ts**2 * (1 - 2 * cot_alpha**2)
```

5. Critical Propagation Angle

```
def critical_angle():
    return math.atan(math.sqrt(2))
```

6. Maximum Growth Rate Value

```
def max_growth_rate(n_s, n, omega_L):
    return 3e-2 * (n_s / n) * omega_L
```

Quantum Circuit Algorithms

1. Optimization Problem Linearization

The objective function from Eq. (3) contains a nonlinear term:

- $q\sqrt{w^\top w}$, which is linearized by introducing a scalar variable `t` and constraint $t \geq \sqrt{w^\top w}$.

Equivalent optimization problem:

min x = (w; t)
$[-\hat{u}; q]^\top (w; t)$

such that:
$1^\top w = 1$,
$|w_i - \bar{w}_i| \leq \zeta_i$,
$w_i \geq 0$,
$t^2 \geq w^\top w$.

2. Second-Order Cone Constraints

Rewriting constraints as second-order cone constraints:

- Let M be an m×n matrix where $=M^\top M$.
- The constraint on t is expressed with an m-dimensional variable η, such that $\eta = Mw$ and:

$(t; \eta) \in Q^{m+1}$.

3. Dual Feasibility and Duality Gap

Given variables:

- Maximize `b⊤y` under constraints:

$A^\top y + s = c$,

$s \in Q$.

Dual feasibility pair:

$(s; y)$ satisfies $A^\top y + s = c$ and $s \in Q$.

Duality gap:

$\mu(x, s) := 1/r\ x^\top s = 1/r\ (c^\top x - b^\top y)$.

4. Absolute-Value Constraints

Handled by introducing variables φ and ρ with equality constraints:

$\varphi = \zeta - (w - \bar{w})$,
$\rho = \zeta + (w - \bar{w})$.

Absolute-value constraints as positivity constraints:

$\varphi_i \geq 0$,
$\rho_i \geq 0$.

Included as second-order cone constraints.

5. Full Optimization Problem

Combining all:

$\min_x\ [-\hat{u}; 0; 0; q; 0]^\top (w; \varphi; \rho; t; \eta) = c^\top x$

such that:

$$\begin{pmatrix} 1^\top & 0^\top & 0^\top & 0 & 0^\top \\ I & I & 0 & 0 & 0 \\ I & 0 & -I & 0 & 0 \\ M & 0 & 0 & 0 & -I \end{pmatrix}$$

$\big)\big(\ w\ \varphi\ \rho\ t\ \eta\ \big)\ |\ |\ |\ \big)\big(\ 1\ \bar{w}+\zeta\ \bar{w}-\zeta\ 0$

By combining these stellar wind algorithms with quantum circuit optimization techniques, we can explore interdisciplinary models and optimize problem-solving approaches across fields.